NAME

tracker3-daemon − Start, stop, restart and list daemons responsible for indexing content

SYNOPSIS

tracker3 daemon [options...]
tracker3 daemon −s | −t [daemons] | −k [daemons] | −l
tracker3 daemon −f | −w [ontology]
tracker3 daemon −−miner <miner> −−pause[−for−process] <reason>
tracker3 daemon −−miner <miner> −−resume <cookie>

DESCRIPTION

Tracker indexes content with daemon processes that run in the background. The **tracker3 daemon** command allows for control of these components. This ranges from starting, stopping and killing processes to pausing and resuming them.

In addition to all this, there are ways to follow or watch what is happening in real time from a top level and right down where the SPARQL commits are happening too.

If no arguments are provided this command will show the current status of all Tracker data miners.

The data miners can be paused or resumed using this command and you can also list miners running and available.

OPTIONS

−p, −−list−processes
This lists all Tracker processes in the system.

*−k, −−kill
This uses SIGKILL to stop all Tracker processes found matching the parameter, if no extra parameter is passed, "all" will be assumed. This is not advised unless you are having problems stopping Tracker in the first place. This **GUARANTEES** death.

*−t, −−terminate=
This uses SIGTERM to stop all Tracker processes. This is recommended over −−kill because it gives the processes time to shutdown cleanly.

−s, −−start
Starts all miners.

−f, −−follow
Follow status changes to daemons as they happen. This is a top level view of what is happening. You will see the name for each daemon and a state with the progress in that state.

This requires Ctrl+C to stop and return to the command line. Each new status is put on a new line.

−w, −−watch=[*ontology***]**
Watch changes that happen to the database in real time. This requires Ctrl+C to stop and return to the command line.

If *ontology* is unspecified, all updates are shown. The *ontology* can be a comma separated list of shorthand or long hand ontology properties. For example:

```
$ tracker3 daemon −w nie:url,nie:mimeType,nfo:fileSize,nie:dataSource
Now listening for resource updates to the database
All nie:plainTextContent properties are omitted

Press Ctrl+C to stop

'nfo:Document'
```

```
    'nfo:fileSize' = '1770'
    'nie:dataSource' = 'http://tracker.api.gnome.org/ontology/v3/tracker#extractor−data−source'
    'nie:mimeType' = 'text/plain'
    'nie:url' = 'file:///home/martyn/.bash_aliases'
  'nfo:Document'
    'nie:dataSource' = 'http://tracker.api.gnome.org/ontology/v3/tracker#extractor−data−source'

  ...
```

−−list−common−statuses
: This will list statuses most commonly produced by miners and the store. These statuses are not translated when sent over D−Bus and should be translated by each application. These are not considered static and are subject to change at any point.

Additionally, these statuses are not the only ones which may be reported by a miner. There may be other states pertaining to the specific roles of the miner in question.

−−list−miners−running
: This will list all miners which have responded to a D−Bus call. Sometimes it is helpful to use this command with **−−list−miners−available.**

−−list−miners−available
: This will list all miners which are available even if they are not running at the moment.

−−pause−details
: For listing all miners which are paused and the reasons for being paused, you can use this. It will also display the application that requested the pause too.

−−miner=<*miner*>
: This argument is used with **−−pause** or **−−resume** to say which miner you want to pause or resume. You can use the full D−Bus name, e.g. "org.freedesktop.Tracker3.Miner.Files" OR you can use the suffix, e.g. "Files".

−−pause=<*reason*>
: The *reason* here is useful to know WHY the miner should be paused. A miner can be paused many times by multiple applications. Only when all pauses have been resumed will it continue. If successful, a cookie will be given to uniquely identify the request. This cookie is used to resume the pause at a later stage.

−−pause−for−process=<*reason*>
: This works exactly the same way as **−−pause** with the exception that it only keeps the pause active while the calling process is alive. As soon as you press Ctrl+C the pause is resumed automatically.

−−resume=<*cookie*>
: The *cookie* is given by a successful **−−pause** command. It is a number which identifies each pause request. When all pauses have been resumed, the miner will resume working.