NAME

tracker-info - Retrieve all information available for a certain file.

SYNOPSIS

tracker info [*options*...] <*file1*> [[*file2*] ...]

DESCRIPTION

tracker info asks for all the known metadata available for the given *file*.

Multiple *file* arguments can be provided to retrieve information about multiple files.

The *file* argument can be either a local path or a URI. It also does not have to be an absolute path.

OPTIONS

-f, --full-namespaces

By default, all keys and values reported about any given *file* are returned in shortened form, for example, *nie:title* is shown instead of *http://www.semanticdesktop.org/ontologies/2007/01/19/nie#title*. This makes things much easier to see generally and the output is less cluttered. This option reverses that so FULL namespaces are shown instead.

-c, --plain-text-content

If the resource being displayed has nie:PlainTextContent (i.e. information about the content of the resource, which could be the contents of a file on the disk), then this option displays that in the output.

-i, --resource-is-iri

In most cases, the *file* argument supplied points to a URL or PATH which is queried for according to the resource associated with it by *nie:url*. However, in cases where the *file* specified turns out to be the actual URN itself, this argument is required to tell "tracker info" not to do the extra step of looking up the URN related by *nie:url*.

For example, consider that you store URNs by the actual URL itself and use the unique nie:url in another resource (which is quite reasonable when using containers and multi-resource conditions), you would need this argument to tell "tracker info" that the *file* supplied is actually a URN not URL.

-t, --turtle

Output results as Turtle RDF. If -f is enabled, full URIs are shown for subjects, predicates and objects; otherwise, shortened URIs are used, and all the prefixes Tracker knows about are printed at the top of the output.

ENVIRONMENT

TRACKER_SPARQL_BACKEND

This option allows you to choose which backend you use for connecting to the database. This choice can limit your functionality. There are three settings.

With "**direct**" the connection to the database is made directly to the file itself on the disk, there is no intermediary daemon or process. The "**direct**" approach is purely *read-only*.

With "**bus**" the **tracker-store** process is used to liase with the database queuing all requests and managing the connections via an IPC / D-Bus. This adds a small overhead *BUT* this is the only approach you can use if you want to *write* to the database.

With "**auto**" the backend is decided for you, much like it would be if this environment variable was undefined.

SEE ALSO

tracker-store(1), **tracker-sparql**(1).

http://nepomuk.semanticdesktop.org/

http://www.w3.org/TR/rdf-sparql-query/