IANCIS
Administrator and User Manual

Usage

Application management

The application is distributed through the file named IANCIS.tar.gz, once the archive is extracted the user should move to the IANCIS directory to start the application. In the IANCIS directory the user can find the scripts used to manage the application, that are described in the following table.

Script name	Description
iancis_start.sh	Start the application
iancis_stop.sh	Stop the application
iancis_reaload_conf.sh	Force the applicaiton to reload the configuration

UNIX signals

The application handles the following POSIX signals: SIGTERM, SIGHUP and SIGINT. If the application receives a SIGINT or SIGTERM it activates the stop procedure, if it receives a SIGHUP it schedules the reload of configuration file (config.properties).

Administration and installation

Prerequisite

Crawling

For crawling operations the application utilizes Tor (https://www.torproject.org) and a HTTP proxy, thus in order to use the application the installation and configuration of these two components is needed. We suggest the use of the privoxy software as HTTP proxy (https://www.privoxy.org).

In its default configuration Tor provides a SOCKS proxy on the port 9050. The HTTP proxy should be configured to use the SOCKS proxy provided by Tor. Privoxy in its default configuration provides a HTTP proxy on the port 8118, in its configuration file must be specified the use of the SOCKS proxy provided by Tor.

Storage

The application uses the ArangoDB (https://www.arangodb.com) database for data storage, thus its installation and configuration is needed for the proper operations of the application.

Configuration

For the application configuration two .properties files are used. The two file are the following:

- crawler.properties
- config.properties

crawler.properties

The file *crawler.properties* contains configuration parameters related to crawler operations. The standard usage of the application shouldn't require changes to this file. In the following we report a short description of the parameters that most likely could be object of changes. For a detailed description of the crawler and of its configuration we refer the reader to the official BUbiNG documentation, see http://law.di.unimi.it/software/bubing-docs/overview-summary.html

rootDir

The name of the directory used by the crawler to store data. This directory is used to store the WARC archive.

parsingThreads

The number of threads used by the crawler to parse web resources.

fetchingThreads

The number of thread used by the crawler to get web resources.

scheduleFilter

Specifies the type of resources to crawl.

parseFilter

Specifies the type of resources to parse to seek new links to follow.

storeFilter

Specifies the type of resources to get and store in the WARC archive.

schemeAuthorityDelay

Specifies the time interval between requests to the same Host.

seed

The file name containing the root set.

proxyHost

The host name ot the http proxy utilized.

proxyPort

The port number of the http proxy utilized.

userAgent

Specifies the user-agetn used while getting web resources.

config.properties

The *config.properties* file contains the configuration parameters of the application. In the following the complete list of configuration parameters and their descripion.

extractorThreads

The number of threads used by the extractor.

analyserThreads

The number of threads used by the analyser.

dbUser

The username used to access the database.

dbPassword

The password used to access the database

dbName

The name of database used to store documents

extractedDocCollection

The name of database's collection used to store extracted text.

analysedDocCollection

The name of database's collection used to store analysed text.

warcSizeThreshold

The threshold size of WARC archive, when reached crawler is stopped and extraction agent is started

crawlerDir

The directory used by crawler to create WARC files.

crawling

Enable/disable crawling operation, possible values are 'true' or 'false'.

extraction

Enable/disable extraction operation, possible values are 'true' or 'false'.

analysis

Enable/disable analysis operation, possible values are 'true' or 'false'.

extractWARC

Enable/disable extraction from WARC, requires 'extraction=true'.

extractFile

Enable/disable extraction from file, requires 'extraction=true'.

fileDir

Directory containing files to extract, requires 'extractFile=true'.

languageFilter

A comma separated list of languages, only texts in these languages are stored and analysed.

storeFilesAnalysis

Enable/disable storage of analysis results on file system, possible values are ‘true’ or ‘false’.

dirFilesAnalysis

Directory used to store analysis results, requires 'storeFilesAnalysis=true'.

storeFilesExtraction

Enable/disable storage of texts extracted on file system, possible values are ‘true’ or ‘false’.

dirFilesExtraction

Directory used to store texts extracted, requires 'storeFilesExtraction=true'.

storeFilesCrawling

Enable/disable storage of crawled data on file system, possible values are ‘true’ or ‘false’.

dirFilesCrawling

Directory used to store crawled data, requires 'storeFilesCrawling=true'.

engineHost

Host name of RESTful web service.

enginePath

Invoked endpoint of RESTful web service.

engineKey

Key used to invoke RESTful web service.

engineKeyFieldName

The name of the form field used to store 'engineKey'