CES Components

Overview

This page provides information about all the components required to build a resilient Concept Extraction API with dynamically updated Gazetteer dictionaries. In case you only need to be able to extract named entities from text with a static dictionary and don't care about high availability, you can do it with a single worker.

Worker

Configuration

General

- -Dworker.name (**OPTIONAL**) - an optional sub-directory within worker persistence directory, points to

  ```
  ~/.ces-worker
  ```

  by default

GATE

- -Dgate.app.location (**REQUIRED**) - full path to the *.xgapp file to load, including the filename and extension. It should start with file:/, otherwise it will be interpreted as relative to the application context
- -Dpipeline-pool-max-size (**OPTIONAL**, default = 1) - the maximum number of Gate pooled applications. In other words, the number of simultaneous annotations this worker will support

Recommended JVM settings

- GC: -XX:+UseConcMarkSweepGC -verbose:gc -verbose:sizes -Xloggc:/path/to/logs/gc.log -XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+PrintTenuringDistribution -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=5 -XX:GCLogFileSize=2M
- Compiler: -XX:+TieredCompilation
- -Xmx: dependent on the pipeline in use, each pipeline package should state how much memory it requires

Coordinator

Configuration

All timeouts are in milliseconds unless specified otherwise.

General

- -Dcoordinator.name (**OPTIONAL**) - the name of this coordinator. Used for suffix for the directory under home in which the coordinator persists its state

- -Dcoordinator.stateDirectory (**OPTIONAL**, default = <home>/.coordinator) - set the directory for Coordinator's state files.
- -Dcoordinator.baseUrl (**REQUIRED**) - the base address of this coordinator. Needed to be able to give workers URLs that point back to the coordinator

GraphDB

- -Dcoordinator.sparql.endpoint (**REQUIRED**) - the remote SPARQL endpoint URL, including repository. Usually in the form http://<host>:<port>/graphdb/repositories/<repo_name>
- -Dcoordinator.sparql.connectionTimeout (**OPTIONAL**, default = 10000) - establish connection to the SPARQL endpoint timeout
- -Dcoordinator.sparql.socketTimeout (**OPTIONAL**, default = 600000) - socket timeout for SPARQL queries

Workers

- -Dcoordinator.worker.connectionTimeout (**OPTIONAL**, default = 10000) - establishing connection to a worker timeout
- -Dcoordinator.worker.socketTimeout (**OPTIONAL**, default = 10000) - socket timeout for worker communication
- -Dcoordinator.worker.retries (**OPTIONAL**, default = 2)
- -Dcoordinator.worker.retryDelay (**OPTIONAL**, default = 2000)
- -Dcoordinator.worker.retryDelayMult (**OPTIONAL**, default = 2.0)

Updates (dictionaries)

- -Dcoordinator.updates.checkDelay (**OPTIONAL**, default = 10000) - initial delay before the first check for updates
- -Dcoordinator.updates.checkRate (**OPTIONAL**, default = 600000) - interval between checks for updates
- -Dcoordinator.updates.maxWorkersToVerify (**OPTIONAL**, default = 2) - a change will first be verified on a single workers before being propagated to all workers. This specified the maximum number of workers to attempt to change before giving up
- -Dcoordinator.updates.verificationTimeout (**OPTIONAL**, default = 1800000) - the maximum time to wait for update verification

Updates (models)

- -Dcoordinator.models.endpoint (**OPTIONAL**) - training node base url. If not specified, worker models won't be updated
- -Dcoordinator.models.schedule (**OPTIONAL**, default = "0 0 2 * * ?") - a cron expression specifying when to check for updates. See [Spring's CronSequenceGenerator documentation](http://docs.spring.io/spring/docs/current/javadoc-api/org/springframework/scheduling/support/CronSequenceGenerator.html) for full syntax and explanation. The default value will check for models every day at 2am.

Annotation

- -Dcoordinator.annotation.freeWorkerTimeout (**OPTIONAL**, default = 30000) - the maximum time to wait for free worker to become available for annotation
- -Dcoordinator.annotation.connectionTimeout (**OPTIONAL**, default = 10000) - establish connection to a worker for annotation timeout
- -Dcoordinator.annotation.socketTimeout (**OPTIONAL**, default = 60000) - socket timeout for annotation to a worker

Watchdog / heartbeat checker

- -Dcoordinator.watchdog.checkDelay (**OPTIONAL**, default = 60000) - initial delay before the first heartbeat check
- -Dcoordinator.watchdog.checkRate (**OPTIONAL**, default = 60000) - interval between heartbeat checks

Files

All files relative to ~/.coordinator/[${coordinator.name}]/ , that is ~/.coordinator if coordinator.name is unset and ~/.coordinator/<coordinator.name>/ if it is set

- workers.json - persisted workers list and configuration
- sparql-update-history.json - the update history for SparqlUpdatesManager
- models.json - latest known models for ModelUpdatesManager

JVM settings

- GC: -XX:+UseConcMarkSweepGC -verbose:gc -verbose:sizes -Xloggc:/path/to/logs/gc.log -XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+PrintTenuringDistribution -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=5 -XX:GCLogFileSize=2M
- Compiler: -XX:+TieredCompilation
- -Xmx: depends on the pipeline, each pipeline should come with memory requirements

GraphDB and EUF plug-in

This is the semantic database you are going to need to enable the dynamic dictionary updates functionality. In case you don't already have GraphDB, go get it here. Official 6.0 documentation.

EUF stands for 'Entity Updates Feed'. This plug-in publishes entity update feeds which are consumed by the Coordinator.

Configuration

To install the EUF plug-in in GraphDB

1. Provide the following Java parameter to GraphDB on startup

   ```
   -Dregister-external-plugins=/your/plugins/home
   ```

2. Unpack the EUF plug-in in your plugins home (prior to starting GraphDB)