cTAKES 4.0 User Install Guide

Contents of this Page

cTAKES 4.0 Links

Apache cTAKES download site

Documentation:

- cTAKES 4.0
- cTAKES 4.0.0.1
- cTAKES 4.0 User Install Guide
- cTAKES 4.0 Developer Install Guide
- cTAKES 4.0 Component Use Guide
- cTAKES 4.0 Dictionaries and Models
- Documentation Conventions

Please see information regarding new UMLS dictionary Authentication at cTAKES 4.0.0.1

These instructions are for end users who want to install Apache cTAKES to process text. If you were planning to expand, change, or modify the code within cTAKES, refer to the cTAKES 4.0 Developer Install Guide.

These instructions will cover installation and running cTAKES against some text. Optional components are described in the Component Use Guide.

Once you have finished installing cTAKES and its separately-bundled resources, you will be able to see what cTAKES is capable of.

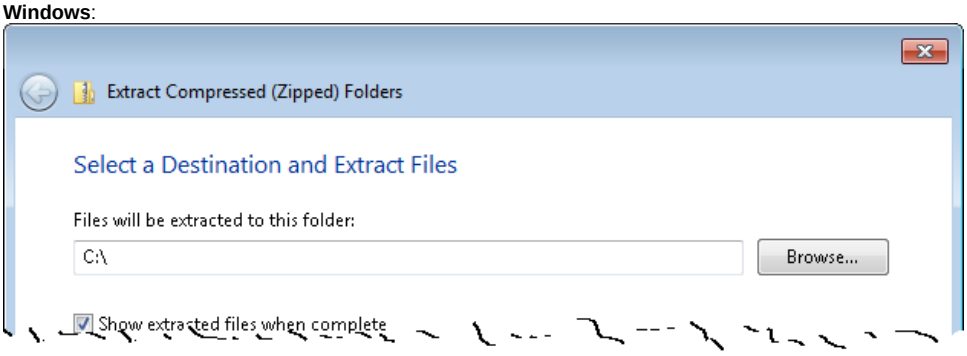
Prerequisites

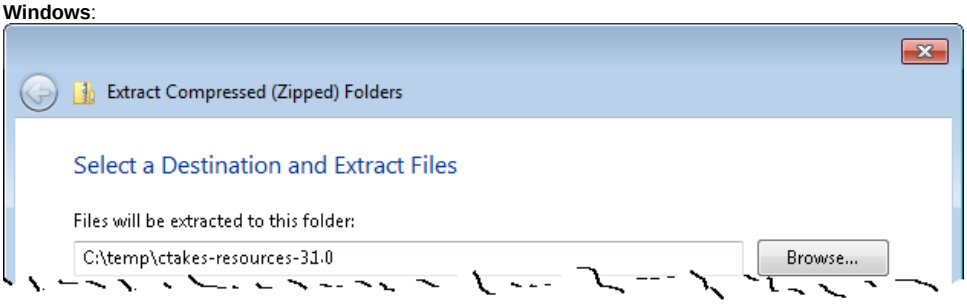
Step	Example

1. Make sure you have Java 1.8 or higher.
 	
Run this command to check your version.
 	
Windows and Linux:
 	
`java -version`
	Windows:
	
	`C:\>java -version`
	`java version "1.8.0_201" Java(TM) SE Runtime Environment ...`
	
	Linux:
	
	`user@system:/$ java -version`
	`java version "1.8.0_201"`
	`OpenJDK Runtime Environment ...`

Install cTAKES

Step	Example
1. On the cTAKES downloads page, download the **User Installation** package. The download time will be commensurate with ~650 MB of data. **404 - page not found** If the download gives you a `404 - page not found` error, go back to the downloads page and select a new `Current Download Mirror` at the bottom of the page.	
2. (Recommended) Verify the downloaded files against a signature to ensure you have the proper and complete file. From the following directory, download the signature file that corresponds to your download from step 1 https://www.apache.org/dist/ctakes/ctakes-4.0.0/ Please do not download any of the files that end with `.zip` or `.gz` directly from a `pache.org/dist` - using the downloads page listed in step 1 for cTAKES install packages will take advantage of using a mirror.	No example

3. Unzip the file you downloaded into a directory that you want to be the cTAKES install location. The compressed files contain a single directory at the top level. This folder we will call <cTAKES_HOME>. It is the directory that contains subdirectories like `bin`, `desc`, `resources`, and `lib`.
 	
You will need to refer to this directory later.
 	
Windows:
 	
`C:\apache-ctakes-4.0.0`
 	
Linux:
 	
`/usr/local/apache-ctakes-4.0.0`
	Windows:
	 
	
	Linux:
	
	`tar -xvf apache-ctakes-4.0.0.bin.tar.gz -C /usr/local`

4. Download the cTAKES resources ZIP file with a matching version from the ctakesresources project (More information on cTAKES models). These resources are required to operate cTAKES.
 	
Due to licensing considerations, resources are hosted at an external location. For ease of installation, a single package was created with all the resources you will need. Licensing for these resources is found within the download.
 	
Download time will be commensurate with 1GB of data.
 	
Unzip the cTAKES resources file into a temporary location.
	Windows:
	 
	
	Linux:
	`cd /tmp`
	`wget http://sourceforge.net/projects/ctakesresources/files/ctakes-resources-4.0-bin.zip`
	`sudo unzip ctakes-resources-4.0-bin.zip`

5. Copy the resources to cTAKES_HOME.
Copy the contents of the temporary resources directory (and all sub-directories) to <cTAKES_HOME> /resources.
 	
There may be conflicts while taking this action. Overwrite the cTAKES_HOME files with those in the resources download.
	Windows:
	`xcopy /s C:\temp\ctakes-resources-4.0-bin\resources C:\apache-ctakes-4.0.0\resources`
	Linux:
	`cp -R /tmp/resources/* /usr/local/apache-ctakes-4.0.0/resources`
	Mac OSX:
	`ditto /tmp/resources/* /usr/local/apache-ctakes-4.0.0/resources`

6. If you created your own dictionaries for use with a previous release of cTAKES and you plan to use them with cTAKES 4.0, you must convert your dictionaries to be compatible with cTAKES 4.0, which is described in the next section. The dictionaries installed by the above steps do not need to be converted.	

Convert Dictionaries You've Previously Created to be Compatible with cTAKES 4.0

cTAKES 4.0.0 uses HSQLDB 2.3.4. Previous version of cTAKES used HSQLDB 1.8. Dictionaries created with HSQLDB 1.8 need to be converted before they can be used by cTAKES 4.0.

Step	Example

1. If you created your own HSQLDB dictionaries for use with a previous release of cTAKES and you plan to use those dictionaries with cTAKES 4.0, you must convert your dictionaries to be compatible with cTAKES 4.0. The dictionaries installed in the preceding section do not need to be converted.	No example

2. If your dictionary's .properties file sets your dictionary's database to be read-only, you need to change it before you can convert it.
 	
• Suggested: make a copy of your database directory for use with 4.0, so that the *filename*.properties and *`filename`*`.script` and any other files in that directory are duplicated, where *filename* is dependent on what you named your database
• Locate the *`filename`*`.properties` file for your database
• Remove these lines, if present:
 	
`readonly=true`
`files_readonly=true`
• Save the *filename*.properties file	

3. Open the database with the 1.8 hsqldb jar:	

Locate the 1.8 hsqldb jar that you used when you created the database. For example, if you used the cTAKES 3.2.2 convenience binary, <cTAKES_HOME_FOR_3.2.2>/lib/hsqldb-1.8.0.10.jar)

If you need to, you can download it from Maven Central at:

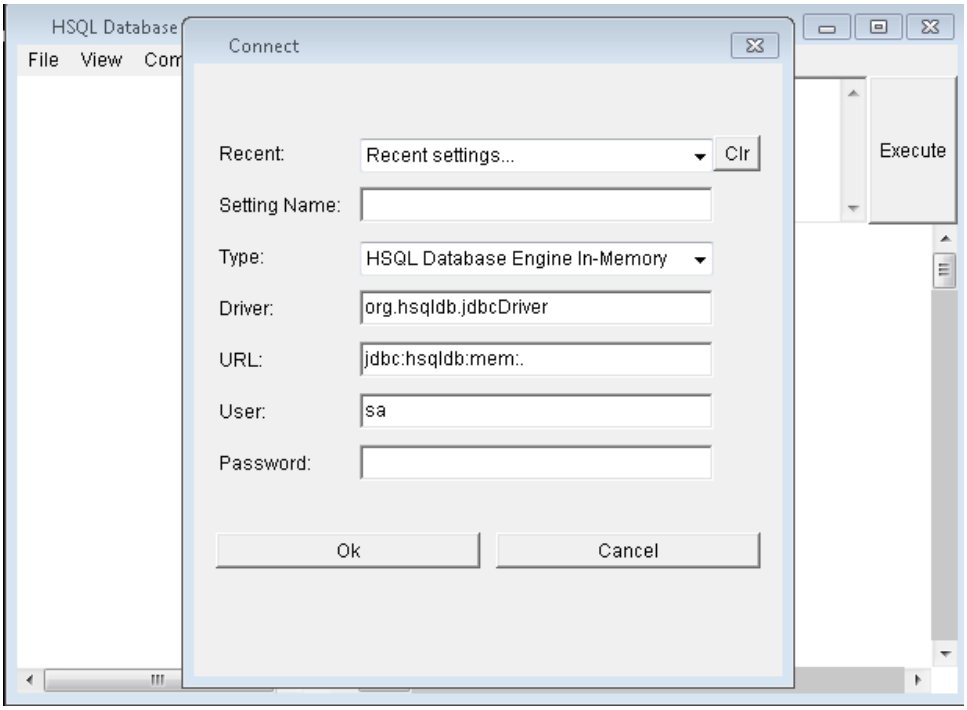
http://central.maven.org/maven2/org/hsqldb/hsqldb/1.8.0.10/hsqldb-1.8.0.10.jar

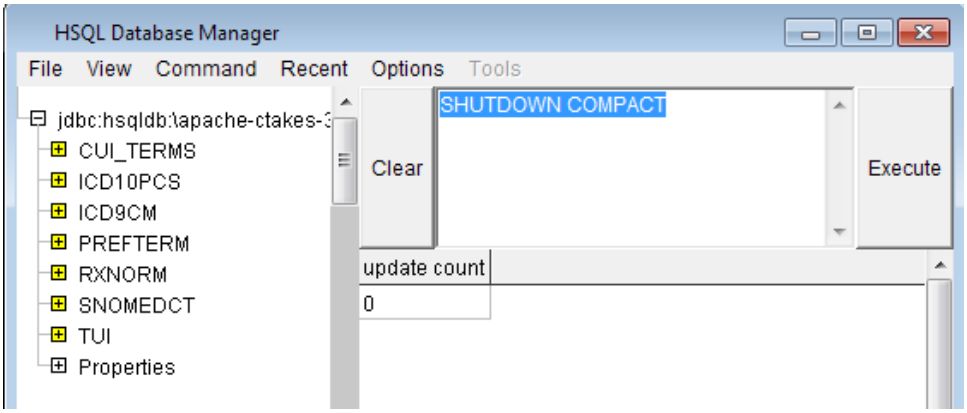
Open the HSQLDB manager GUI for version 1.8. For example, if your 1.8 jar is in `C:\Apps\hsqldb\`, you would enter this command:

```
java -cp C:\Apps\hsqldb\hsqldb-1.8.0.10.jar org.hsqldb.util.DatabaseManager
```

Connect to your database, by entering the appropriate URL and pressing the Ok button.

For example, if you are on Windows and your dictionary's `.properties` file is



`C:\cTAKES_3\resources\org\apache\ctakes\dictionary\lookup\fast\customdict\`**`custom`**`.properties` you could enter the following for the URL `jdbc:hsqldb:\cTAKES_3\resources\org\apache\ctakes\dictionary\lookup\fast\customdict\`**`custom`**	
4. Using HSQLDB 1.8, in the upper right pane, enter `SET SCRIPTFORMAT TEXT` and press the `Execute` button. After the update count appears, go to the next step.	
5. Using HSQLDB 1.8, in the upper right pane, enter `SHUTDOWN COMPACT` and press the `Execute` button. After the update count appears, exit the Database Manager GUI.	

6. Now do the same with hsqldb 2.3.4 jar - open the HSQLDB 2.3.4 database manager GUI: `java -cp C:\apache-ctakes-4.0.0\lib\`**`hsqldb-2.3.4.jar`** `org.hsqldb.util.DatabaseManager` Connect to your database, by entering the appropriate URL and pressing the Ok button. In the upper right pane, enter `SHUTDOWN COMPACT` and press the `Execute` button. After the update count appears, exit the Database Manager GUI.	
7. Verify the *filename*.properties file for your database contains `version=2.3.4` If it doesn't, make sure the .properties file does not have read only=true the .properties file does not have `files_readonly=true` you used `hsqldb-`**`2.3.4`**`.jar` when instructed to	

8. Suggested: Set your dictionary's database to be read-only, by adding `read only=true` to the *filename*.properties file.	
9. Repeat the above steps for each of your dictionaries that you had created for use with a previous release of cTAKES.	

(Recommended) Add UMLS access rights

> ⚠ In the initial setup cTAKES will recognize only few sample concepts in text. If you wish to perform named entity recognition or concept identification for anything other than these few words, you will need to 1) obtain the rights to use UMLS resources 2) add those credentials to cTAKES, and 3) use a cTAKES pipeline that makes use of those UMLS resources. If you don't, cTAKES will work but won't recognize much.

Step	Example
1. If you do not have a UMLS username and password, you may request one at UMLS Terminology Services.	No example

2. Once you have your UMLS username and password, edit the following files. Find the lines in each script that runs java and add the ctakes.umlsuser and ctakes.umlspw parameters to the java command with your credentials. Make sure you substitute your actual ID and password if you cut and paste the example.

Windows:

```
<cTAKES_HOME>\bin\runctakesCVD.bat
<cTAKES_HOME>\bin\runctakesCPE.bat
```

Linux:

```
<cTAKES_HOME>/bin/runctakesCVD.sh
<cTAKES_HOME>/bin/runctakesCPE.sh
```

In the examples below, the rest of the lines after `-cp` are not shown because you do not need to modify the rest of the line. Do not delete the rest of the line after `-cp` however.

```
java -Dctakes.
umlsuser=<YOUR_UMLS_ID_HERE> -Dctakes.
umlspw=<YOUR_UMLS_PASSSWORD_HERE> -cp
...
```

If you use special characters in your user name or password, you may need to escape them or for windows, place the string in quotes

For example, if your username and password were literally myusername and mypassword, you could insert them before the -cp option so the start of the java command would look like this:

```
java  -Dctakes.umlsuser=myusername  -
Dctakes.umlspw=mypassword  -cp ...
```

Windows:

If you use special characters in your umls user name or password, you can place them in double-quotes:

```
java  -Dctakes.umlsuser="
myuser!!!!"  -Dctakes.umlspw="
mypass!!!!"  -cp ...
```

The rest of the line after `-cp` is not shown because you do not need to modify the rest of the line. Do not delete the rest of the line after `-cp` however.

Linux:

If you use special characters in your user name or password, you may need to escape them

2a. You may also specify your UMLS Credentials as environment variables to your operating system, but the dots will need to be replaced with underscores.

Windows:

```
REM this sets it for the current
command window
set  ctakes_umlsuser=YourUmlsUserId
set  ctakes_umlspw=YourUmlsPassword
```

Linux:

```
export ctakes_umlsuser=myusername
export ctakes_umlspw=mypassword
```

Process documents using cTAKES

This version allows you to test most components bundled in cTAKES in the following ways:

1. Using the bundled UIMA CAS Visual Debugger (CVD) to run a pipeline and view the results. Also allows you to view results that have been saved as XCAS files

2. Using the bundled UIMA Collection Processing Engine (CPE) to process documents in a directory and save the results in another directory.
3. Using the cTAKES 4.0 Simple Pipeline Fabricator GUI

On Linux, you will need a windowing environment to run these tools.

CAS Visual Debugger (CVD)

Step	Example

1. Open a command prompt and change to the cTAKES_HOME directory, which is the directory that contains subdirectories like bin, desc, resources, lib.
 	
Depending on how you extracted the files,
 	
It is best if <cTAKES_HOME> is your current directory. The scripts will change directories, so being home to run the command is best.
	Windows:
	
	`cd \apache-ctakes-4.0.0`
	`-- or --`
	`cd \apache-ctakes-4.0.0-bin\apache-ctakes-4.0.0\`
	
	Linux:
	
	`cd /usr/local/apache-ctakes-4.0.0`
	`-- or --`
	`cd /usr/local/apache-ctakes-4.0.0-bin/apache-ctakes-4.0.0`

2. This step uses AggregatePlaintextFastUMLSProcessor, which requires that you downloaded the cTAKES resources as described in step 4 of Install cTAKES (above) and that you added UMLS access rights (also above).
 	
If you haven't done those, you can use the AggregatePlaintextProcessor instead.
 	
Start the CAS Visual Debugger and load the AggregatePlaintextFastUMLSProcessor pipeline by running this command (at right)
 	
The application may take a minute to start on slower hardware.
 	
The GUI opens and then loads the AggregatePlaintextFastUMLSProcessor pipeline. If it appears to be hung, look at the window where you entered the command and you will see what is happening.
 	
Once the analysis engine has successfully loaded you should see a tree in the **Analysis Results frame**:
 	
CAS Index Repository
** SofaIndex [0]*
** AnnotationIndex [1]*
	Windows:
	
	`bin\runctakesCVD.bat desc\ctakes-clinical-pipeline\desc\analysis_engine\AggregatePlaintextFastUMLSProcessor.xml`
	
	Linux:
	
	`bin/runctakesCVD.sh -desc desc/ctakes-clinical-pipeline/desc/analysis_engine/AggregatePlaintextFastUMLSProcessor.xml`

3. Copy the example text from the next cell in this table and paste the contents into the Text section of CVD, replacing the text that is already there. You can also download a copy of the file from here	`Dr. Nutritious Medical Nutrition Therapy for Hyperlipidemia Referral from: Julie Tester, RD, LD, CNSD Phone contact: (555) 555-1212 Height: 144 cm Current Weight: 45 kg Date of current weight: 02-29-2001 Admit Weight: 53 kg BMI: 18 kg/m2 Diet: General Daily Calorie needs (kcals): 1500 calories, assessed as HB + 20% for activity. Daily Protein needs: 40 grams, assessed as 1.0 g/kg. Pt has been on a 3-day calorie count and has had an average intake of 1100 calories. She was instructed to drink 2-3 cans of liquid supplement to help promote weight gain. She agrees with the plan and has my number for further assessment. May want a Resting Metabolic Rate as well. She takes an aspirin a day for knee pain.`

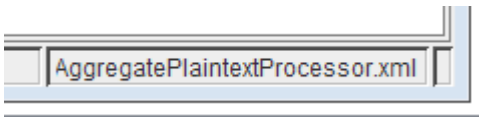
4. From the menu bar, click **Run** -> **Run AggregatePlaintextFastUMLSProcessor**. Note: If you would like to TEST some simple annotators to ensure it's working without UMLS, you can just load: /desc/ctakes-core/desc/analysis_egine /SentencesAndTokensAggregate.xml	
5. You'll get a list of all the annotations for this clinical document in the Analysis Results frame. Annotations such as named entities, division by sentence, etc from the pipeline are viewable. To see one, in the **Analysis Results** frame, click on the key in front of: > *CAS Index Repository* > ** AnnotationIndex* > ** uima.tcas.Annotation* > ** org.apache.ctakes. typsystem.type.textsem. IdentifiedAnnotation* > ** org.apache.ctakes. typsystem.type.textsem. EventMention* This will show an AnnotationIndex in the lower frame. Select any annotation in that lower frame and you will see the text discovered in the text frame on the right like the concept of the disease/disorder Hyperlipidemia. For a medication example select this > *CAS Index Repository* > ** AnnotationIndex* > ** uima.tcas.Annotation* > ** org.apache.ctakes. typsystem.type.textsem. IdentifiedAnnotation* > ** org.apache.ctakes. typsystem.type.textsem. EventMention* > ** org.apache.ctakes. typsystem.type.textsem. MedicationMention* Now select items in the lower frame to see the text being annotated.	

To run other pipelines, use the **Run**-> **Load AE** menu bar command.

Navigate to the file you wish to load, such as

```
<cTAKES_HOME>
  /desc
    /ctakes-clinical-pipeline
      /desc
        /analysis_engine
          /AggregatePlaintextProcessor.xml
```

Click **Open**.
Loading the analysis engine may take a minute. The lower right corner of the window shows the name of the currently-loaded pipeline if a pipeline was loaded successfully.



You may close the **CAS Visual Debugger (CVD)** application if you wish.

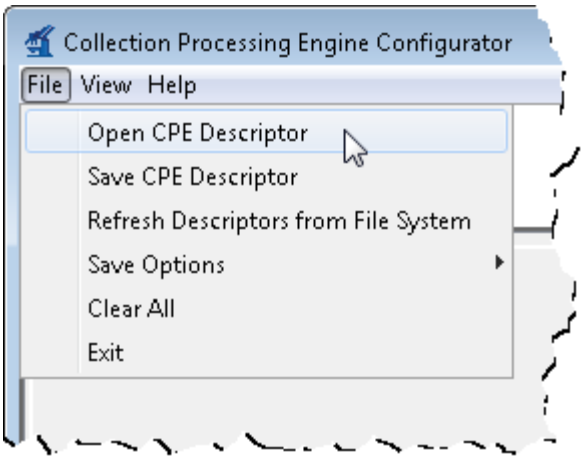
Collection Processing Engine (CPE)

Step	Example

1. Open a command prompt and change to the cTAKES_HOME directory, which is the directory that contains subdirectories like bin, desc, resources, lib.
 	
It is best if <cTAKES_HOME> is your current directory. The scripts will change directories, so being home to run the command is best.
	Windows:
	`cd \apache-ctakes-4.0.0`
	
	Linux:
	`cd /usr/local/apache-ctakes-4.0.0`

2. Create a directory for some test data.	`mkdir testdata`
3. The sample dictionary that does not require UMLS rights contains only a few terms. Create a file containing the sentence at right into the testdata directory.	The patient says they took 325 mg aspirin for knee pain.

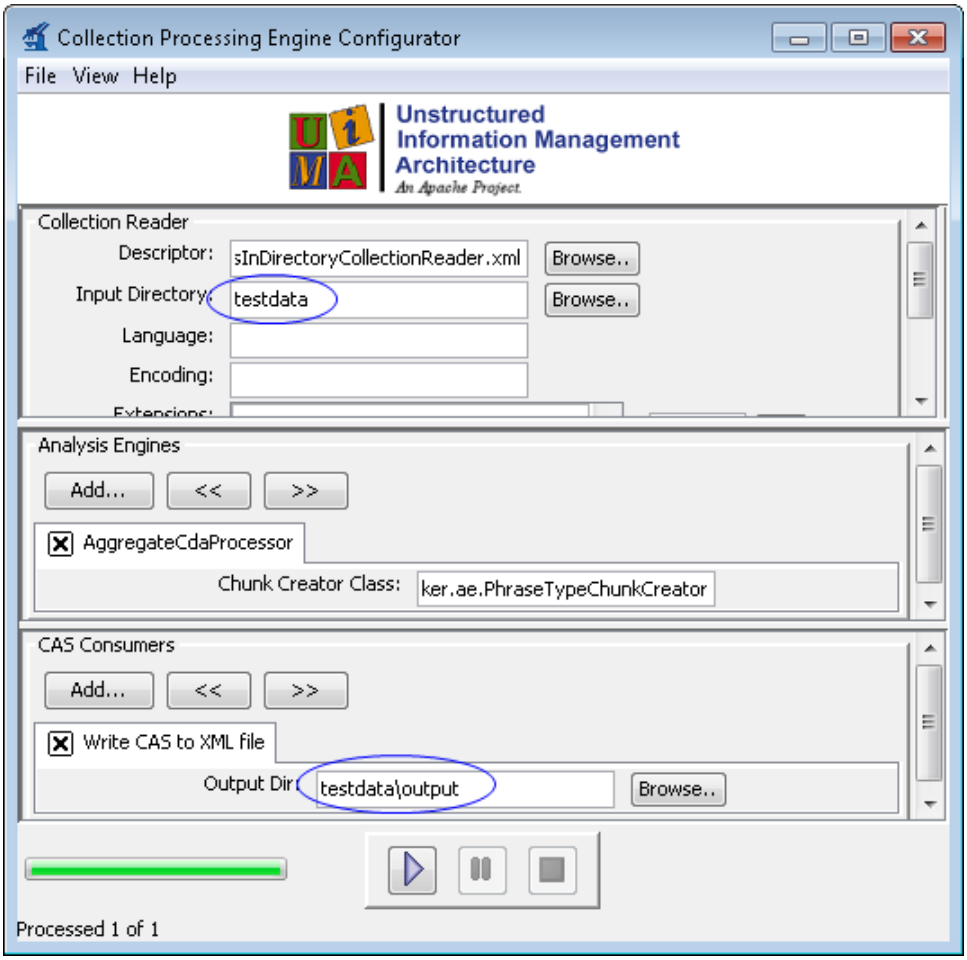
4. Start the collection processing engine by running this command:
The application may take a minute to start on slower hardware.
	Windows:
	`bin\runctakesCPE.bat`
	
	Linux:
	`bin/runctakesCPE.sh`

5. This will bring up the Collection Processing Engine Configurator. In the Menu bar click **File** >**Open CPE Descriptor**	

6. Navigate to the following file, which uses the AggregatePlaintextProcessor
 	
`<cTAKES_HOME>`
`/desc`
`/ctakes-clinical-pipeline`
`/desc`
`/collection_processing_engine`
`/test_plaintext.xml`
 	
Click **Open**.
	No example

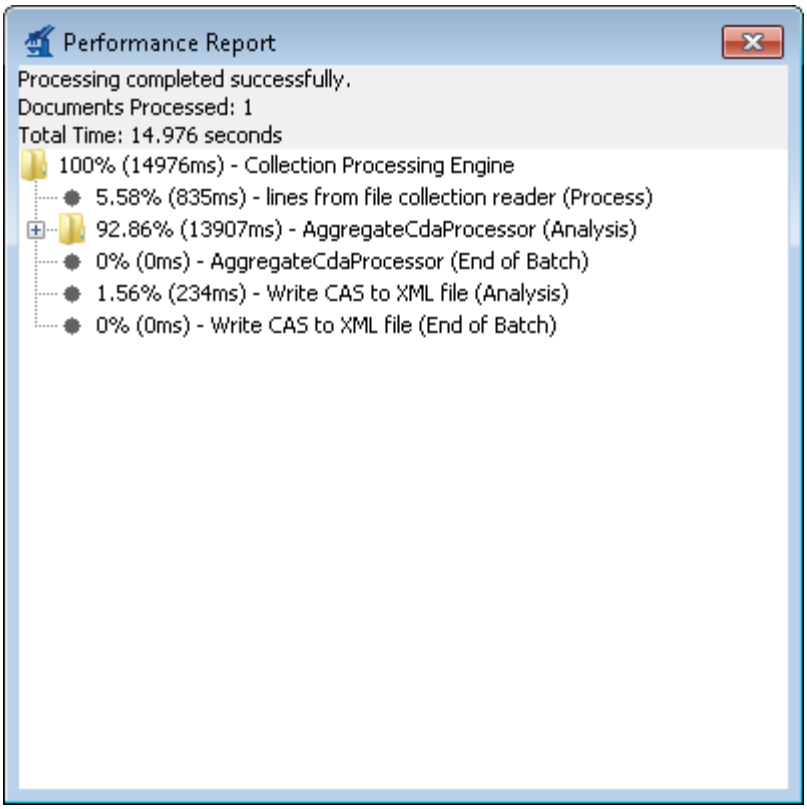
7. Change the Collection Reader input directory to testdata, which contains the files to process

Within the CAS Consumers pane of the same window, change the output directory to testdata/output



8. Click the Play button (green/blue **play arrow** near the bottom).

What just happened? The pipeline used a file system reader that will process all files in a directory. The processing was accomplished by a pipeline of cTAKES components. For each input file, one resultant file was placed into the output directory. Each output file is an XML file that includes the annotations made by each component within the pipeline. (The AggregatePlaintextProcessor allows for the Chunk Creator Class parameter to be passed to the Chunker annotator.)

9. You should see that one document was processed. You did process a collection of documents. In this case the collection only contained one just to show how to do it. Close the Performance Report window. This example of using the CPE GUI did not use the UMLS resources. If you wish to perform named entity recognition or concept identification for anything other than a few words, you will need to 1) obtain the rights to use UMLS resources 2) add those credentials to cTAKES, and 3) use a pipeline that makes use of those UMLS resources (see above).	
10. Close the CPE application. You may be prompted to save changes. Since this was just a test you may click the **No** button.	No example

cTAKES Pipeline Fabricator GUI (Creating Piper Files)

The cTAKES GUI can be launched using bin\runPiperCreator.bat or bin\runPiperCreator.sh

Step 1: Open a command prompt and change to the cTAKES_HOME directory, which is the directory that contains subdirectories like bin, desc, resources, lib.

Step 2 for Windows: bin\runPiperCreator.bat

Step 2 for Linux: bin\runPiperCreator.sh

Step 3: Allow the GUI to scan for annotators

Step 4: Select which elements to include in your pipeline

Step 5: (Recommended) Save your pipeline definition

Step 6: Run the pipeline using the Run icon 

Step 7: Examine your output.

Analysis Engines/Pipelines

The analysis engines shipped with cTAKES for some of the annotators are described in the following table.

Annotator	Description	Example Piper file
Clinical Pipeline	The pipeline to obtain concepts and their attributes	<cTAKES_HOME>\resources\org\apache\ctakes\clinical\pipeline\DefaultFastPipeline.piper
Chunker	Obtains phrasal chunk annotations	<cTAKES_HOME>/TBD

Dependency Parser	Obtains dependency parsing tree	<cTAKES_HOME>/TBD
Drug NER	Finds mentions of medications and medication attributes such as dose, strength, frequency...	<cTAKES_HOME>/TBD
Dictionary Lookup	Finds mentions of concepts from a dictionary (e.g., SNOMED CT or RxNorm	<cTAKES_HOME>/TBD
Dictionary Lookup Fast	Finds mentions of concepts from a dictionary (e.g., SNOMED CT or RxNorm	<cTAKES_HOME>/TBD
Relation Extractor	Finds certain relations (location of and degree of) between certain Event, Entity, and Modifier annotations	<cTAKES_HOME>/TBD
Smoking Status	Finds document or patient-level smoking status	<cTAKES_HOME>/TBD
Side Effect	Finds side effect mentions and sentences from clinical documents	<cTAKES_HOME>/TBD

Next Steps

To run cTAKES from a command line, see Default Clinical Pipeline.

The cTAKES 4.0 Component Use Guide will help you to understand each of the cTAKES components that have been installed. In some cases you can learn how to improve the components.

Also, before you go on to process text in production, you will want to consider dictionaries and models. If you did not obtain the rights yet to the UMLS resources and models, you will want to do so. Be aware, the models within cTAKES have been trained on data that may not match your data well enough to be effective. In some cases you might want to create your own dictionaries, or modify the dictionaries and train models using your own data.