

cTAKES 4.0 Developer Install Guide

Contents of this Page

- [Please see information regarding new UMLS dictionary Authentication at cTAKES 4.0.0.1](#)

About

Getting the code

- [Subversion](#)

Build

IDE based installs

- [Eclipse minimal install instructions](#)
- [Eclipse step by step install instructions](#)
- [IntelliJ IDEA step by step install instructions](#)
 - [Perform a fresh checkout from SVN](#)
 - [Import Project from Maven](#)
 - [Compile with Maven](#)
 - [Running UIMA CVD with Maven](#)

Testing

- [Running UTests with UMLS credentials](#)
- [Release Manager](#)

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](#)

About

These are instructions for installation of cTAKES for developers. With these instructions you can set up your development environment with cTAKES code so you can compile the code and change or extend it. If you simply want to be a user of the software, refer to the [cTAKES 4.0 User Install Guide](#).

These install instructions do not supply knowledge of what the cTAKES components do. That is found in the [cTAKES 4.0 Component Use Guide](#) and the pages for the individual components.

Getting the code

Subversion

Subversion is the default VCS used by cTAKES.

Project PATH

There is current issue with the development setup where there are *hardcodings* to the level the project is supposed to exist relative to M2_HOME (e.g. ~/ .m2) folder.

It is recommended to checkout/clone 3 levels from M2_HOME. (e.g. ~/projects/apache/ctakes). We will name this folder as CTAKES_DEV_HOME

```
export CTAKES_DEV_HOME=~/.projects/ctakes
```

```
cd ${CTAKES_DEV_HOME}
svn checkout https://svn.apache.org/repos/asf/ctakes/trunk
export CTAKES_SRC_HOME=${CTAKES_DEV_HOME}/trunk
cd ${CTAKES_SRC_HOME}
```

Build

The standard Maven command for building cTAKES is:

```
$ mvn clean package -DskipTests=true
```

If you would like to build with the same settings as the default Jenkins job (recommended only for committers), use:

```
$ rm -fr /tmp/ctakes-*
$ exports MAVEN_OPTS='-Xms3072m -Xmx4g -Xss128M -XX:+CMSClassUnloadingEnabled -XX:-UseGCOverheadLimit -Dmaven.test.failure.ignore=false'
$ mvn --fail-at-end --errors --update-snapshots clean install sonar:sonar -DskipTests=false -Dsonar.scm.provider=svn -Dsonar.host.url=https://builds.apache.org/analysis
```

IDE based installs

Below are instructions for Eclipse and for IntelliJ

Eclipse minimal install instructions

Prerequisites: Java JDK 1.8+

If you already have Eclipse installed, ensure you have [Apache Maven](#) or equivalent IDE plugin, Subclipse plugin or [Apache Subversion](#) or equivalent IDE plugin.



The location below is the main trunk of cTAKES. See <https://svn.apache.org/repos/asf/ctakes/> for branches and tags.

- Import Project > Maven > Checkout Maven Project from SCM
 - use: svn and <https://svn.apache.org/repos/asf/ctakes/trunk>
 - Select the option to Check out Head Revision.
 - Select the option to Check out All Projects.
- Wait until Eclipse downloads and builds all of your projects (it may take up to 30 minutes or more depending on your internet connection and your hardware).
 - The various build helpers should run jcasgen and build the projects for you. There should not be any reason to run mvn install, etc.
- If you are using the Fast Dictionary Lookup, the dictionary will be downloaded for you when you do a maven build.
- If you want to use the full LVG or original dictionary lookup, merge resources from <https://sourceforge.net/projects/ctakesresources/files/ctakes-resources-4.0-bin.zip/download> into the relevant subproject.
- (Optional) If you would like to launch the UIMA CVD or CPE GUI, run `ctakes-clinical-pipeline/resources/launch/UIMA_<CVD | CPE>GUI--clinical_documents_pipeline.launch`

- (Optional) UIMA plug-ins called "UIMA Eclipse tooling and runtime support" can be installed from update site: <http://www.apache.org/dist/uima/eclipse-update-site>

Eclipse step by step install instructions

- These steps assume a Windows or Ubuntu Linux install environment. You will need to extrapolate for any other environments.
- These steps were written using Mars aka Eclipse 4.5. You may need to adapt some steps for other versions of Eclipse.
- Recommend searching for "Eclipse IDE for Java Developers" and downloading the desired version
- Within Eclipse, install the Maven SCM connector
 - File -> Import -> Maven -> Check out Maven Projects from SCM
 - Click Next.
 - Click the "m2e Marketplace" link (somewhat above the Cancel button)
 - You are now in the m2e Marketplace not the Eclipse Marketplace.
 - Scroll to and select m2e-subversive
 - Click Next, Next, the radio button for Accept, Finish
 - restart Eclipse when prompted
- Within Eclipse, install the Subversive Connector
 - Again, use File -> Import -> Maven -> Check out Maven Projects from SCM
 - The Install Connectors window will open.
 - Select one of them and install it: (On Windows 7 Pro, "**Native JavaHL 1.8.14 (64 bit)**" works
 - Click Finish
 - Accept the install by clicking: Next, Next, the radio button for Accept, Finish.
 - When prompted to restart Eclipse, Click Yes
- You can set your UMLS userID and password to be passed to all Java programs you launch within Eclipse, using Windows->Preferences->Java->Installed JREs->(highlight the JRE you are using)->Edit->Default VM arguments:
 - Dctakes.umluser=YourUmlsUserIdGoesHere "-Dctakes.umlspw=YourUmlsPasswordGoesHere"
- Check out cTAKES:
 - For a third time, Import-> Maven -> Check out Maven Projects from SCM
 - For SCM URL, select **svn** and in the text field enter: <https://svn.apache.org/repos/asf/ctakes/trunk>
 - Click Finish. Eclipse will download all of the cTAKES sub-projects

 Due to the way Maven and Eclipse work together you will see two copies of the sub-projects in Eclipse. If you look into your workspace directories there is only one set of underlying files.

- If you don't have Eclipse set to Build Automatically, you'll need perform a build:
 - Select all projects, right click, and choose Maven->Update Projects. Click Select All, and click OK to start the maven build.
 - You can view the Progress pane by using the Eclipse menu option Window->Show View->Progress
 - Performing a maven build includes running jcasgen as needed.
- The dictionary for the Fast Dictionary Lookup was downloaded as part of the Maven build.
- If you want to use the full LVG or original dictionary lookup, merge resources from <https://sourceforge.net/projects/ctakesresources/files/ctakes-resources-4.0-bin.zip/download> into the relevant subproject.
- (Optional) UIMA plug-ins called "UIMA Eclipse tooling and runtime support" can be installed from update site: <http://www.apache.org/dist/uima/eclipse-update-site>

IntelliJ IDEA step by step install instructions

Disclaimer: The specific process may change in any IDE between versions.

Perform a fresh checkout from SVN

1. Start IntelliJ IDEa.
2. In the "Quick Start" menu, select "Check out from Version Control". This will display a drop-down box.

3. In the drop-down box, select "Subversion".
This will open a "Checkout from Subversion" dialog.
4. In the "Checkout from Subversion" dialog, click the "+" button in the top left to add a new Repository.
This will open a "New Repository Location" dialog.
5. In the "New Repository Location" dialog, enter the svn checkout location of cTakes.
6. Click "Ok".
This will inspect the repository.
7. Click the "Expand" triangle.
This should display the directory listing of trunk.
8. Click "Checkout".
This will open a "Destination Directory" dialog.
9. Enter a local directory in which to keep trunk (your sandbox).
10. Click "Ok". This will open a "Checkout Options" dialog.
The default options ("Head", etc.) are fine for most users.
11. Click "Ok".
This will open a "Working Copy Format" dialog.
12. Select a (version) format and click "Ok". I use version 1.8, but any should be fine.
This will start the actual checkout and display a progress dialog. The checkout may take a little while.
13. After the checkout has completed, a new dialog will ask you if you'd like to open the project. Click "No".

Import Project from Maven

1. Start IntelliJ IDEa.
2. In the "Quick Start" menu, select "Import Project".
This will open a "Select File or Directory to Import" dialog.
3. Browse to your local cTakes repository root directory and select the pom.xml file.
4. Click "Ok".
This will open an "Import Project from Maven" dialog.
5. Make sure the "Search for projects recursively" box is selected, just in case any cTakes modules are not in the pom.
6. Make sure that "Create IntelliJ IDEa modules for aggregator projects" is not selected.
If you plan to add new module, 'disable' a present cTakes module or make other changes to the main pom.xml, check the "Import Maven projects automatically" box.
See also: <http://www.jetbrains.com/idea/webhelp/maven-importing.html>
7. Make sure that "Create module rgoups for multi-module Maven projects" is not selected.
8. Make sure that "Keep source and test folders on reimport is selected.
9. Make sure that "Exclude build directory (%PROJECT_ROOT%/target)" is selected.
10. Make sure that "Use Maven output directories" is selected.
11. Make sure that the "Generated sources folders" option "Detect automatically" is selected.
12. For the "Phase to be used for folders update" the default option "process-resources" should be fine.
13. For the "Automatically download" options, you may select what you like, but be wary that if broken code has been checked in you may need to revert manually.
14. The default "Dependency types" are fine.
15. Click "Environment settings...".
This will open a dialog that can be used to set options about the Maven environment.
16. The default maven environment settings should be fine. If \$M2_HOME is not set in your environment you may select a "Maven home directory", but it is better to set \$M2_HOME in your environment.
17. Click "Next".
This will inspect the cTakes Maven settings and search for profiles. It should display a dialog with the possible cTakes profiles.
If you plan to run the UIMA CVD or CPE then select the appropriate profile. Neither is necessary
18. Click "Next".
This will open an "Import" dialog with the current version of cTakes displayed.
19. Click "Next".
This will open a dialog allowing you to select a Java SDK version.
20. Click the "+" button in the top left.
This will display a drop-down box with options for an SDK.
21. Select "JDK".
This will open a dialog to select a Java JDK directory.
22. Navigate to a directory with a JDK version 1.8 or later and click "Ok".
This will display a listing of the file paths associated with the selected JDK.
23. Click "Next".
24. Click "Ok/Next".
The project will load. This may take a while.
25. Important: If you are asked about adding any .iml files to svn, click "No".
26. You should now see the full cTakes project structure in IntelliJ.

Compile with Maven

1. Open the "Maven Projects" Tool Window using the button on the left side of the window.
If you do not see it, Use the main Menu > View > Tool Windows > Maven Projects.
If you would like to permanently add the button to the UI, use Menu > View > Tool Buttons.
See also: <http://www.jetbrains.com/idea/webhelp/maven-projects-tool-window.html>
2. Select the "Expand" triangle next to "Apache cTAKES" and then its child node "Lifecycle".
This will display standard maven goals.
3. Right-click on "compile".
This will display a selection of available actions.
4. Select "Create 'ctakes [compile]'"

This will open a dialog with options for the Maven "compile" goal.

5. Make sure "Profiles (...)" is empty.

6. Click "Ok".

7. You should now see a new child of "Apache cTakes" named "Run Configurations", with the child node "ctakes [compile]"

8. Right-click on "ctakes [compile]" and select "Run" from the popup menu.

9. A panel should appear at the bottom of the screen and display the progress of the compilation. This may take a while.

Running UIMA CVD with Maven

1. Open the "Maven Projects" Tool Window using the button on the left side of the window.

If you do not see it, Use the main Menu > View > Tool Windows > Maven Projects.

If you would like to permanently add the button to the UI, use Menu > View > Tool Buttons.

See also: <http://www.jetbrains.com/idea/webhelp/maven-projects-tool-window.html>

2. Select the "Expand" triangle next to "Apache cTAKES" and then its child node "Lifecycle".

This will display standard maven goals.

3. Right-click on "compile".

This will display a selection of available actions.

4. Select "Create 'ctakes [compile]'"

This will open a dialog with options for the Maven "compile" goal.

5. In the "Name" box at the top of the dialog type "cTakes runCVD".

6. In "Profiles (...)", type "runCVD".

7. Click "Ok".

8. You should now see a new child of "Apache cTakes" named "Run Configurations", with the child node "cTakes runCVD"

9. Right-click on "cTakes runCVD" and select "Run" from the popup menu.

10. A panel should appear at the bottom of the screen and display the progress of the compilation. This may take a while.

11. The UIMA CVD should launch when the compilation is complete.

12. All standard output and error from the CVD will be piped to the IntelliJ Window.

Testing

Running UTests with UMLS credentials

UMLS tests are executed if the `ctakes_umlsuser` environment variable is defined.

On Windows:

```
> set ctakes_umlsuser=<your user>
> set ctakes_umlspw=<your password>
> mvn clean test
> REM or running oneline command:
> mvn -Dctakes_umlsuser=<your user> -Dctakes_umlspw=<your password> clean test
```

On Linux:

```
$ export ctakes_umlsuser=<your user>
$ export ctakes_umlspw=<your password>
$ mvn clean test
$ # or running oneline command:
$ mvn -Dctakes_umlsuser=<your user> -Dctakes_umlspw=<your password> clean test
```

Release Manager

The release manager will use the `apache-release` profile defined in the parent pom. For this profile, the UMLS UTests are enforced.