

# UIMA Installation

- **Apache UIMA Framework Download:**

UIMA Java framework & SDK (Binary) von dieser Seite runterladen und in ein geeignetes Verzeichnis entpacken:

<http://uima.apache.org/downloads.cgi#Latest> Official Releases

- **Environment Variables** (aus dem apache-uima README):

- Set JAVA\_HOME to the directory of your JRE installation you would like to use for UIMA.
- Set UIMA\_HOME to the apache-uima directory of your unpacked Apache UIMA distribution.
- Append UIMA\_HOME/bin to your PATH.
- Please run the script UIMA\_HOME/bin/adjustExamplePaths.bat (or .sh), to update paths in the examples based on the actual UIMA\_HOME directory path. This script runs a Java program; you must either have java in your PATH or set the environment variable JAVA\_HOME to a suitable JRE.

**Anmerkung:** Mein JAVA\_HOME Path zeigt auf *JDK*, *nicht JRE*!

(Konkret: C:\Program Files\Java\jdk1.8.0.92)

- **Eclipse Tools einrichten** (siehe apache-uima/docs/d/overview\_and\_setup):

In Eclipse: **Help** → **Install New Software...**

In das Eingabefeld neben *work with:* <http://www.apache.org/dist/uima/eclipse-update-site/> und mit Enter bestätigen. Jetzt die Plugin Tools auswählen und installieren. (Ich habe alle 3 Optionen installiert).

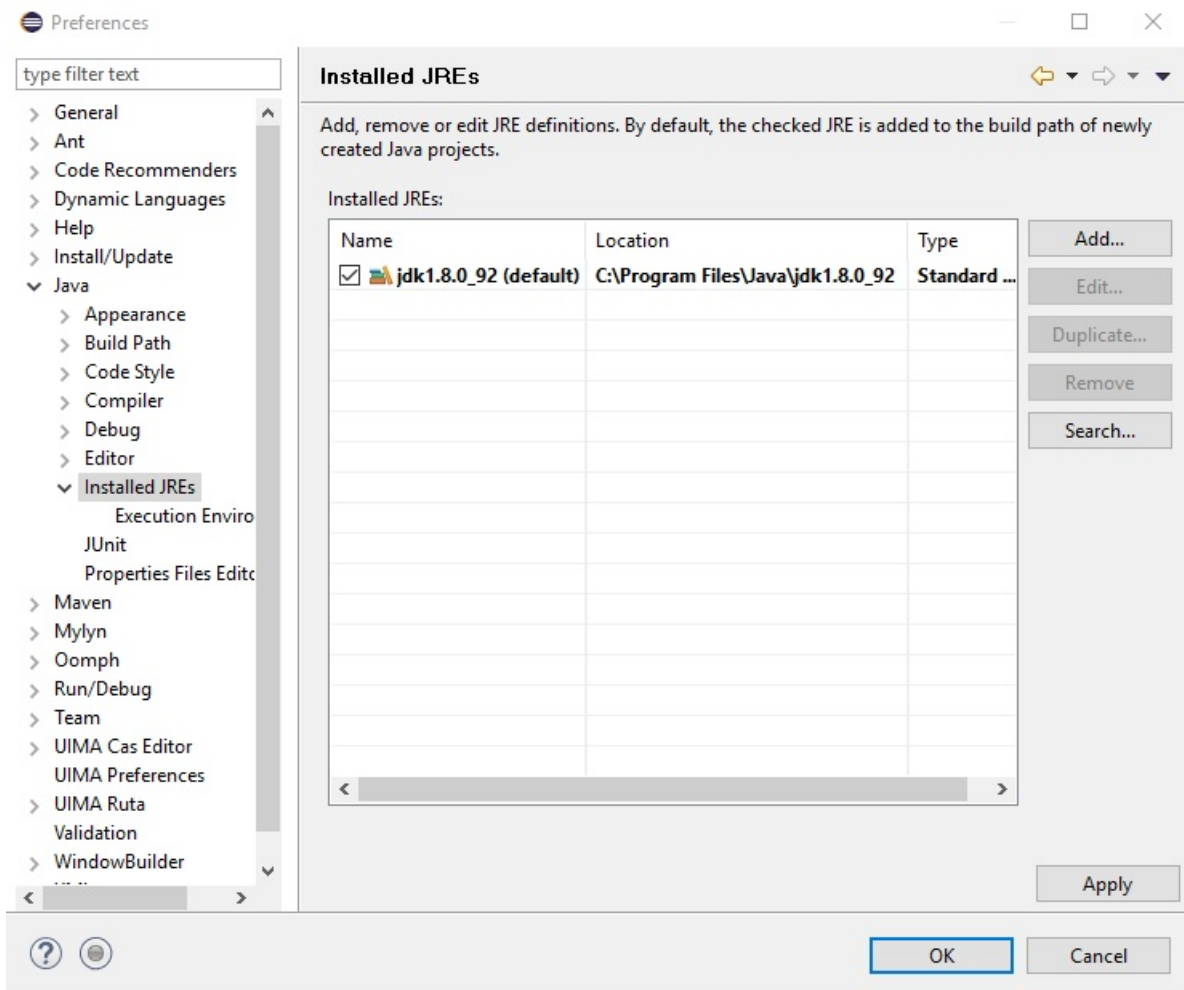
Eclipse neustarten.

**Anmerkung:** Mit Eclipse Neon funktioniert dieser Schritt nicht! Mars oder frühere Version benutzen. (Stand: 30.06.2016).

- **Eclipse für Example Code konfigurieren:**

- **Installed JREs überprüfen:**

- \* In Eclipse: Window → Preferences → Java → Installed JREs  
Hier muss als Standard ein JDK angegeben sein, kein JRE!



Falls es anders aussieht, ändern!

#### – UIMA\_HOME classpath einstellen:

- \* In Eclipse: Window → Preferences → Java → Build Path → Classpath Variables
- \* Auf *New* klicken.
- \* In das Feld *Name* UIMA\_HOME eingeben.
- \* Installationsverzeichnis bei *Path* angeben (Bsp: C:/Program Files/apache-uima).
- \* mit *OK* bestätigen.

#### – Beispiel-Projekt importieren:

- \* File → Import
- \* *General/Existing Project into Workspace* auswählen und dann *next*.
- \* *Browse* und zum %UIMA\_HOME% - Verzeichnis navigieren.
- \* *Finish* wählen. In Eclipse sollte nun das Projekt *uimaj-examples* ohne Kompilierungsfehler sein.

- **UIMA source code zu den jar files hinzufügen:**

Aus overview and setup 3.3:

**Note:** If you are running a current version of Eclipse, and have the m2e (Maven extensions for Eclipse) plugin installed, Eclipse should be able to automatically download the source for the jars, so you may not need to do anything special (it does take a few seconds, and you need an internet connection).

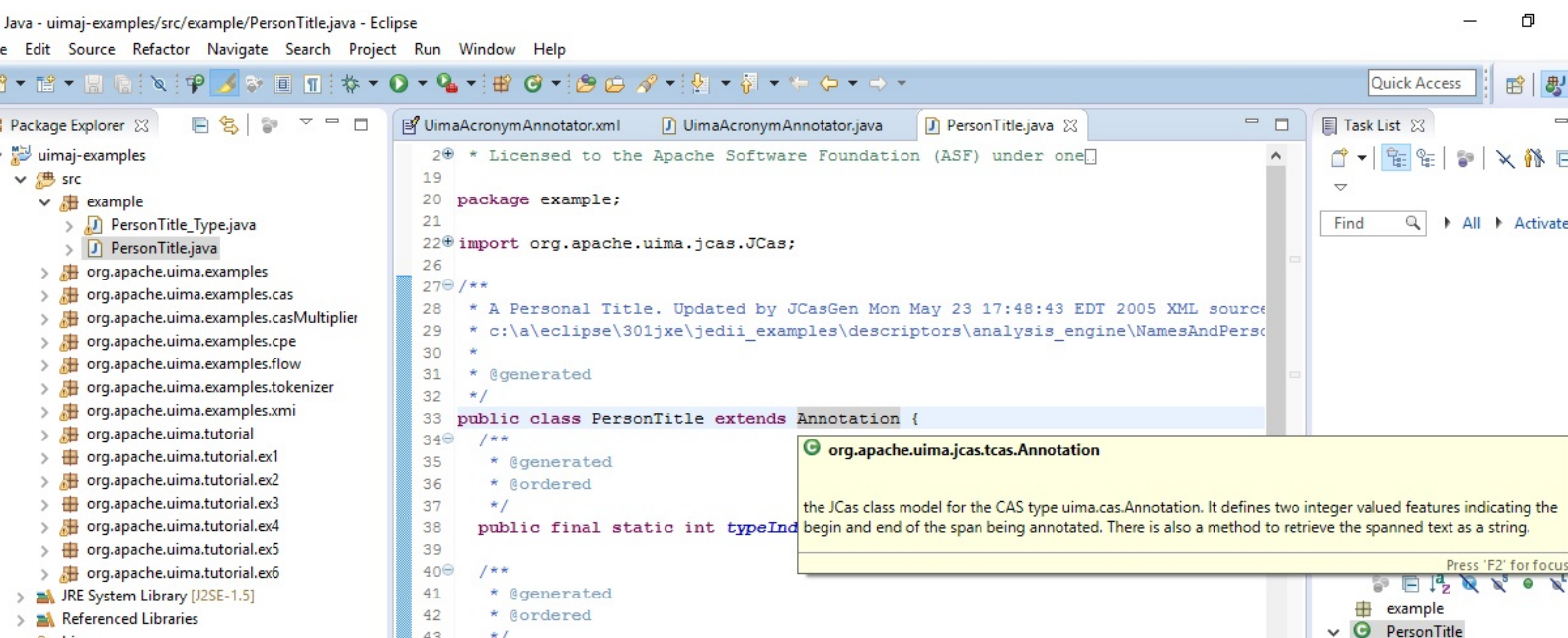
Wenn dies nicht der Fall ist, den manuellen Weg wählen:

- UIMA source distribution runterladen.  
<http://uima.apache.org/downloads.cgi#Latest> Official Releases  
(Diesmal in der Spalte Source statt Binary)
- Source-Distribution entpacken (Ordnername: uimaj-2.8.1).
- Im Verzeichnis uimaj-2.8.1\src\main\readme\_src liegt das Script addSourceToJars.

Dies darf *nicht* an Ort und Stelle ausgeführt werden, sondern muss in dem root-Verzeichnis der Source-Distribution verwendet werden.

Sprich: addSourceToJars muss direkt in uimaj-2.8.1 ausgeführt werden.

Ob es richtig funktioniert hat zeigt sich, wenn man in Eclipse im uimaj-examples-Projekt eine Klasse wie PersonTitle.java öffnet und die Maus z.B. über Annotation bewegt. Wenn die Sources in den Jars sind, sollte jetzt auch die Java-Doc angezeigt werden:



Wenn alles geklappt hat, sollten jetzt auch die Tools von Eclipse aus ausführbar sein.

- Run → Run... sollte viele Optionen, wie z.B. AnnotationViewerMain liefern. Die Tools sollten ohne Fehlermeldung starten.
- Alternativ auch im Package Explorer Rechtsklick auf das Projekt *uimaj-examples* → Run As → Java Application → Tool wie beispielsweise AnnotationViewerMain wählen. Es sollte sich ein kleines AnnotationViewer-Fenster öffnen.

Wenn etwas nicht geklappt hat, gibt es noch diesen Schritt aus dem README zu beachten:

### **Building from the Source Distribution**

We use Maven 3.0 or later for building; download this if needed, and set the environment variable MAVEN\_OPTS to -Xmx800m -XX:MaxPerSize=256m.

Then do the build by going into the .../uimaj directory, and issuing the command `mvn clean install`

This builds everything except the ...source-release.zip file. If you want that, change the command to

```
mvn clean install -Papache-release
```

Look for the result here: `target/uimaj-[version]-source-release.zip` (if run with `-Papache-release`)

For more details, please see <http://uima.apache.org/building-uima.html>

- Ein kurzes Anwendungstutorial gibt es hier:  
<http://uima.apache.org/doc-uima-annotator.html>
- Ein ausführliches Tutorial befindet sich in `apache-uima/docs/d/tutorials_and_users_guides`