

UIMA sample Analysis Engine



Agenda

- Install UIMA SDK
- Eclipse setup
- Extract UIMA sample project
- Cas Visual Debugger (CVD) tool usage
- Setup own UIMA annotator project
- Test with CVD
- Create Pear file
- Test pear file

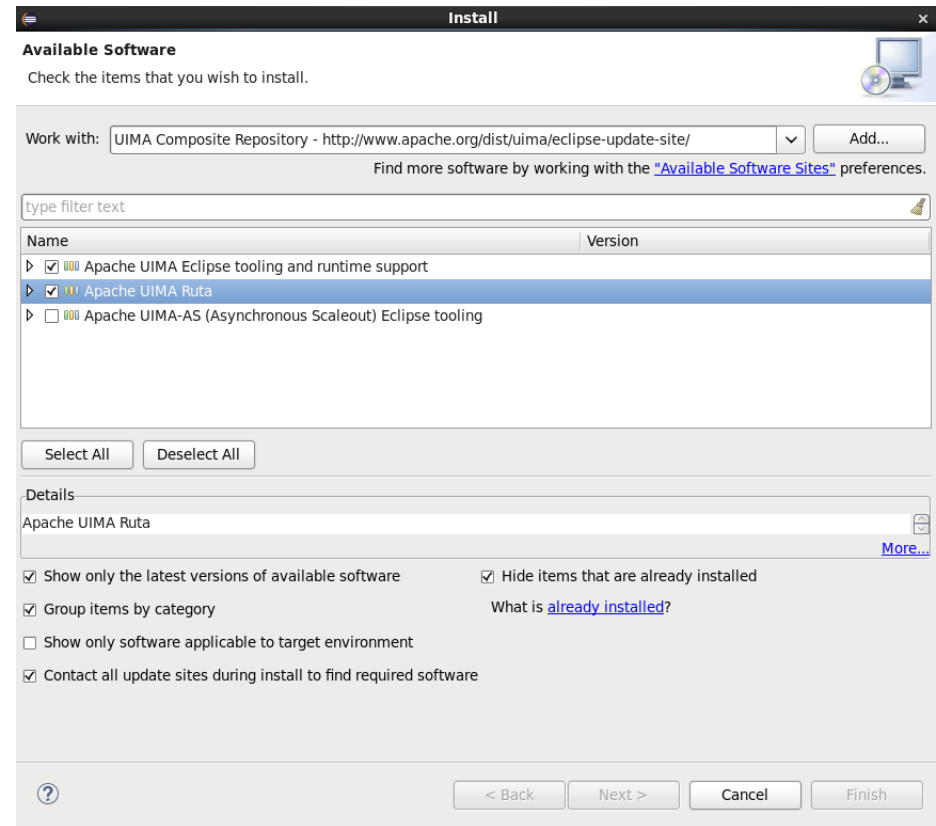
Install UIMA SDK

- Download binary from www-eu.apache.org/dist/uima/uimaj-2.8.1/uimaj-2.8.1-bin.tar.gz
- Download sources from www-eu.apache.org/dist/uima/uimaj-2.8.1/uimaj-2.8.1-source-release.zip
- Unpack e.g. in /home/streamsadmin/UIMA
- Adapt environment

```
# add for UIMA
export UIMA_HOME=/home/streamsadmin/UIMA/apache-uima
export PATH=$UIMA_HOME/bin:$PATH
export TREETAGGER_HOME=/home/streamsadmin/UIMA-Ruta/tree-tagger-linux-3.2.1
```
- Check that JAVA_HOME is already set
- Run `adjustExamplePaths.sh`

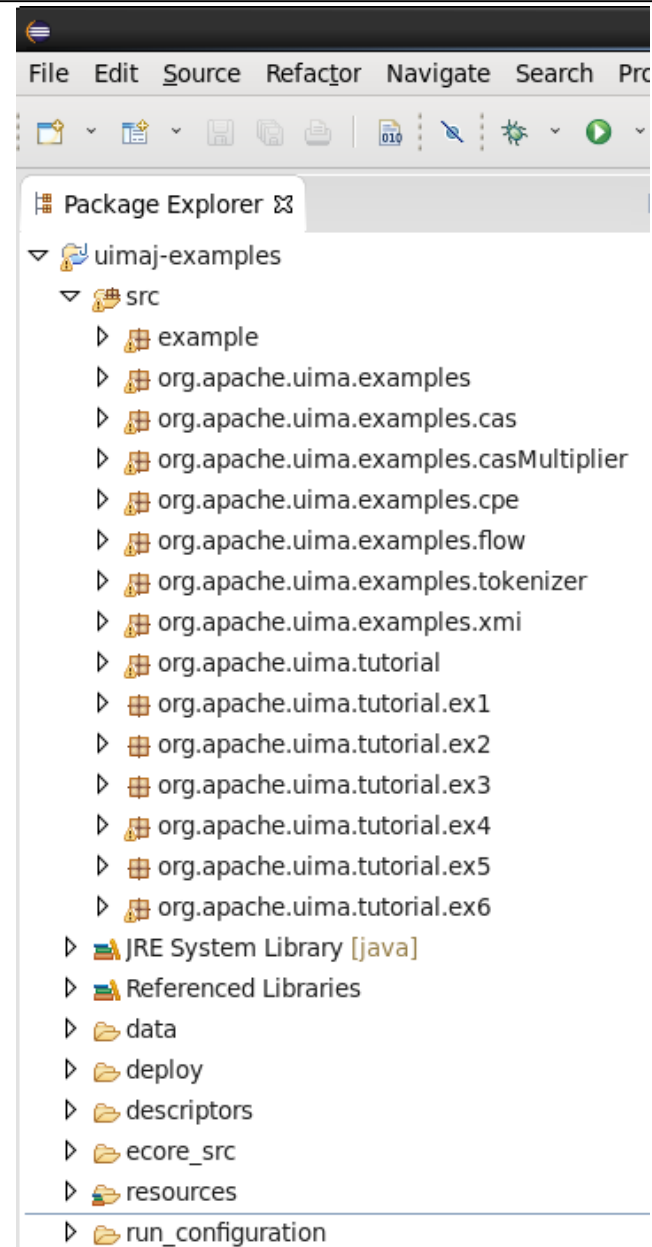
UIMA Eclipse plugin installation

- Get Eclipse Luna (for Java) from eclipse.org.
- Start Eclipse
- Install UIMS Ruta plugin:
Help->Install new software -> Add
<http://www.apache.org/dist/uima/eclipse-update-site/> -> select “Apache UIMA Eclipse tooling and runtime support”, install
- Restart Eclipse



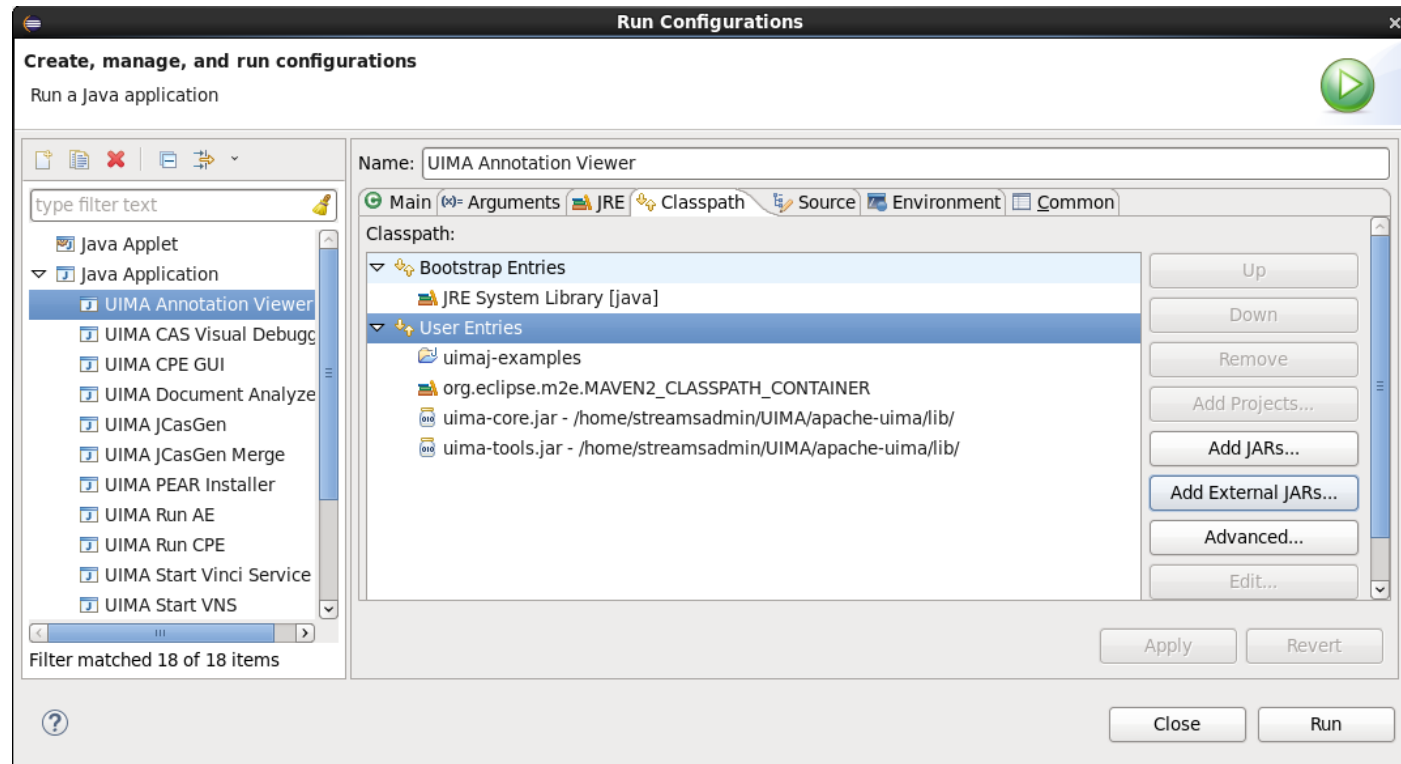
Load UIMA sample projects

- In the eclipse Java perspective
Eclipse -> Window -> Open perspective -> Java
- Set up a class path variable named UIMA_HOME
Eclipse -> Window -> Preferences -> Java -> Build Path -> Classpath Variables -> New
Name: UIMA_HOME
Path: /home/streamsadmin/UIMA/apache-uima
- Import sample project
Eclipse -> File -> Import -> General -> Existing Project into Workspace -> Next
Browse to find UIMA SDK install root dir (e.g. /home/streamsadmin/UIMA/apache-uima)
It will show Projects: **uimaj-examples** (/home/streamsadmin/UIMA/apache-uima/examples)
Finish



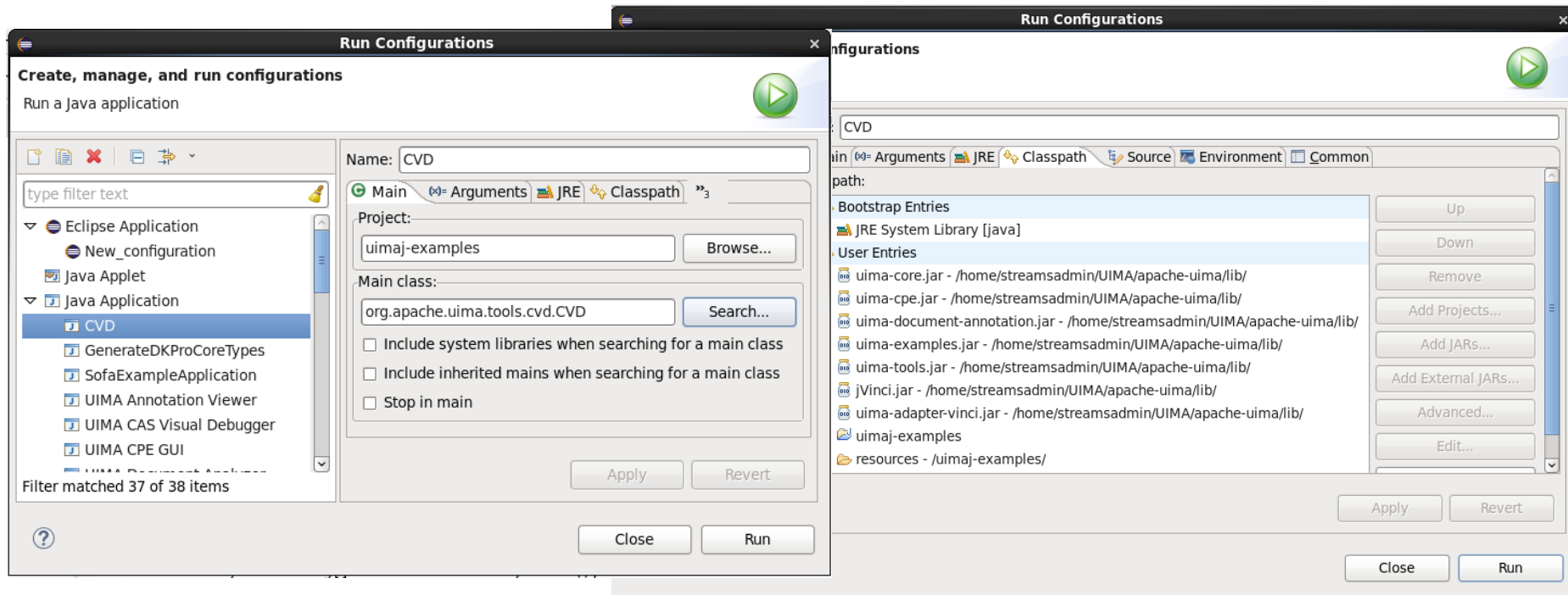
Run external tools from Eclipse

- Eclipse -> Run -> Run Configurations...
in the left navigation: Java Application -> UIMA *
per UIMA tool: select classpath tab -> Add external Jars... -> add uima-core.jar and uima-tools.jar
(e.g. from /home/streamsadmin/UIMA/apache-uima/lib)
Apply, Run
- To run a tool with a certain project, This project needs to be added to the classpath as well (like e.g. uimaj-examples)



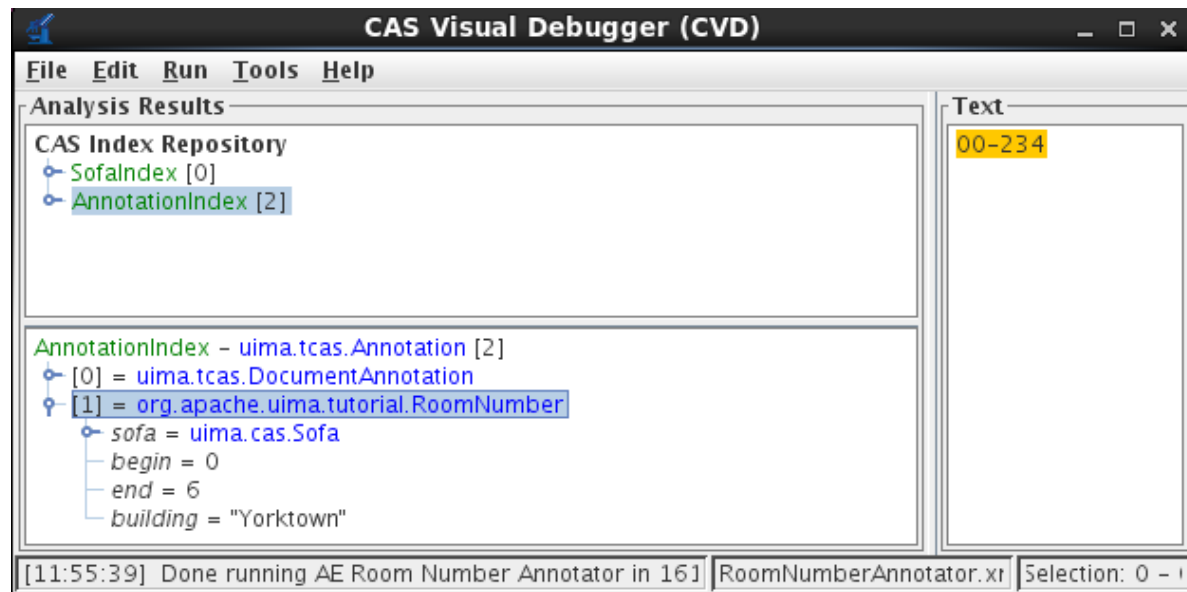
Run CVD from Eclipse with the uimaj-example projects

- Eclipse -> Run -> Run Configurations...
 - Duplicate UIMA CAS Visual Debugger Run Config and name it CVD
 - Main tab: Enter your project (select via Browse)
 - Arguments tab: empty arguments
 - Classpath tab: in User Entries
 - Add Projects... uimaj-examples
 - (if class not found : Advanced -> Add Folder... add bin folder of the project)
 - Add External JARs... see screen shot for examples
 - Advanced -> Add Folder... resources (from uimaj-examples)
 - Apply, Run



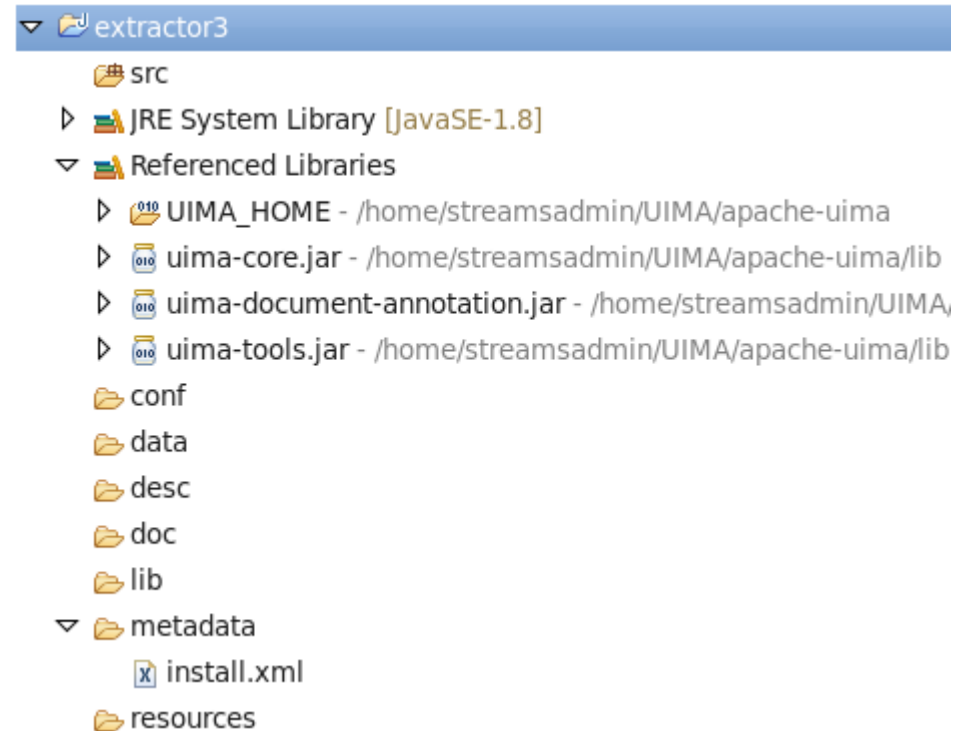
CVD – CAS Visual Debugger

- CVD -> Run -> Load AE
 - /home/streamsadmin/workspace/uimaj-examples/descriptors/tutorial/ex1/RoomNumberAnnotator.xml
- In the Text area enter some text
 - containing a room number format like 00-123
- CVD -> Run -> Run Room Number Annotator
- In the Analysis result area select AnnotationIndex
 - In the below area select org.apache.uima.tutorial.RoomNumber
 - That should cause the room number in the Text area to get background color



Setup an own UIMA project

- Eclipse -> File -> New -> Java Project
 - Project Name: e.g. extractor3
 - Next, select libraries tab
 - Add external jars...
 - uima-core.jar (e.g. from /home/streamsadmin/UIMA/apache-uima/lib)
 - uima-document-annotation.jar (e.g. from /home/streamsadmin/UIMA/apache-uima/lib)
 - uima-tools.jar (e.g. from /home/streamsadmin/UIMA/apache-uima/lib)
 - Add variable...
 - UIMA_HOME - /home/streamsadmin/UIMA/apache-uima
 - Finish
- On the new Project (via context menu): Add UIMA Nature#
 - That creates a UIMA directory structure under the project



Setup own UIMA project (cont)

- In the new projects desc folder (extractor3/desc) create an AE descriptor
 - Eclipse -> File -> New -> Other... -> Analysis Engine Descriptor File..., Finish
- Open the new aeDescriptor.xml (via context menu)
 - Open with... -> Component Descriptor Editor
 - In the Overview tab: enter the name of the planned Java class file (e.g. com.ibm.extractor3.KeyPhraseAnnotator)
 - In the Type Sytem tab: Add planned CAS types (e.g. Add Type
 - Type Name: com.ibm.extractor3.KeyPhrase, Supertype: uima.tcas.Annotation)
 - Optional add additional feature attributes to the type (e.g. Add..
 - Feature Name: Category, Range Type: uima.cas.String
 - Feature Name: CategoryID, Range Type: uima.cas.Integer)
 - Save the aeDescriptor.xml changes (ignore the error message for now)
 - In the Type System Tab press button: JCasGen (generates package and java files under src folder)

The screenshot shows the Eclipse IDE with a project named 'extractor3' selected in the Package Explorer. The project structure includes a 'src' folder containing 'com.ibm.extractor3' with files 'KeyPhrase_Type.java' and 'KeyPhrase.java'. Other folders like 'JRE System Library', 'Referenced Libraries', 'conf', 'data', 'desc' (containing 'aeDescriptor.xml'), 'doc', 'lib', 'metadata', and 'resources' are also visible.

The 'Type System Definition' dialog is open, showing the 'Types (or Classes)' tab. It lists the following types:

Type Name or Feature Name	SuperType or Range
com.ibm.extractor3.KeyPhrase	uima.tcas.Annotation
Category	uima.cas.String
CategoryID	uima.cas.Integer

Buttons on the right include 'Add Type', 'Add...', 'Edit...', 'Remove', 'Export...', 'JCasGen', and a 'limited' checkbox. At the bottom, tabs for 'Overview', 'Aggregate', 'Parameters', 'Parameter Settings', 'Type System', and 'Capabiliti' are visible.

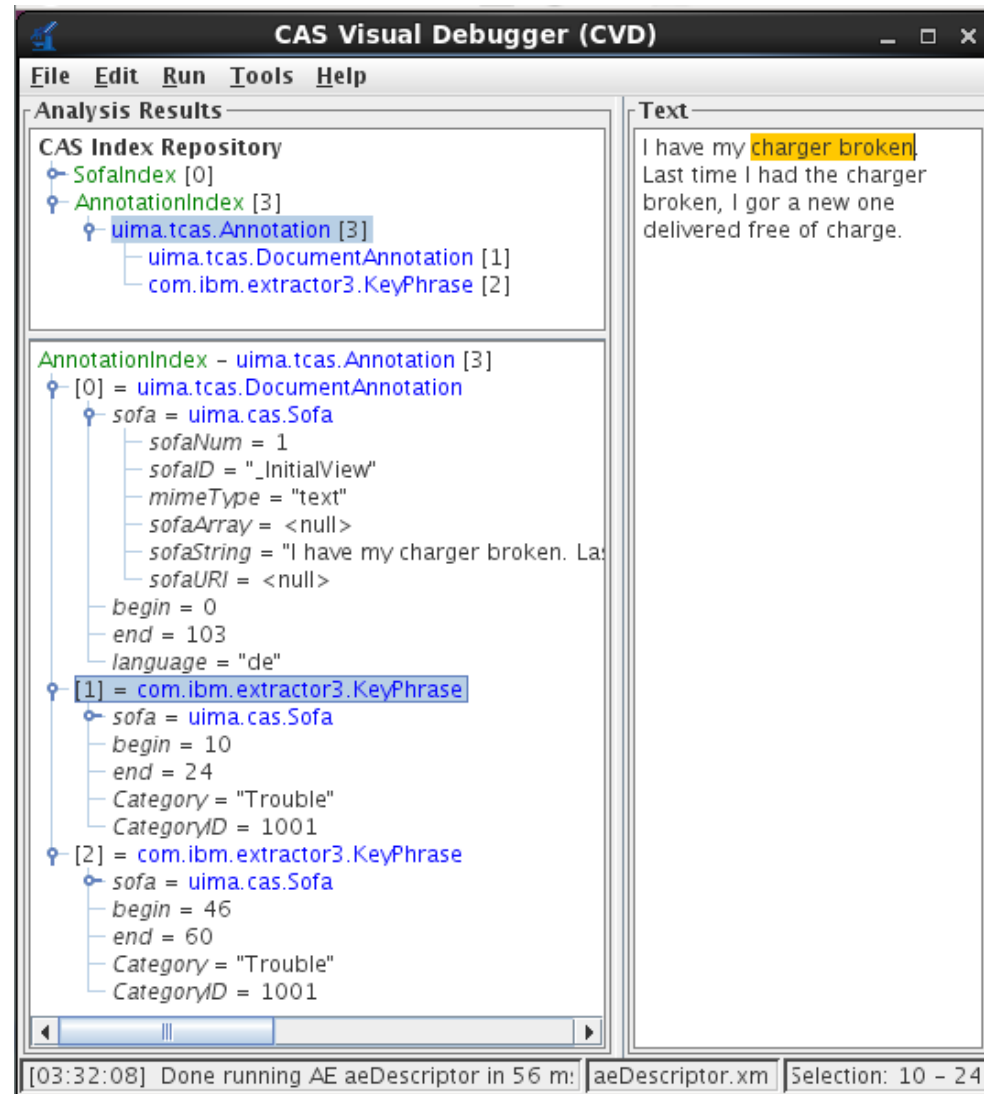
Setup own UIMA project (cont)

- In the generated package folder (extractor3/src/com.ibm.extractor3) create the class
 - Eclipse -> File -> New -> Class
 - Name (as previously defined in the aeDescriptor.xml): e.g. KeyPhraseAnnotator
 - Superclass: org.apache.uima.analysis_component.JCasAnnotator_ImplBase
 - Check to create method stubs for inherited abstract methods, Finish
- Open the generated file and add your java code to the process method (this example extracts “charger broken” from a text, in reality we would rather read the pattern(s) from a resource file)

```
package com.ibm.extractor3;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import org.apache.uima.analysis_component.JCasAnnotator_ImplBase;
import org.apache.uima.analysis_engine.AnalysisEngineProcessException;
import org.apache.uima.jcas.JCas;
import com.ibm.extractor3.KeyPhrase;
public class KeyPhraseAnnotator extends JCasAnnotator_ImplBase {
    @Override
    public void process(JCas arg0) throws AnalysisEngineProcessException {
        String text = arg0.getDocumentText();
        text = text.toLowerCase();
        Pattern p = Pattern.compile("charger broken", Pattern.CASE_INSENSITIVE+Pattern.LITERAL);
        Matcher m = p.matcher(text);
        int pos = 0;
        while (m.find(pos)) {
            KeyPhrase annotation = new KeyPhrase(arg0);
            annotation.setBegin(m.start());
            annotation.setEnd(m.end());
            annotation.setCategory("Trouble");
            annotation.setCategoryID(1001);
            annotation.addToIndexes();
            pos = m.end();
        }
    }
}
```

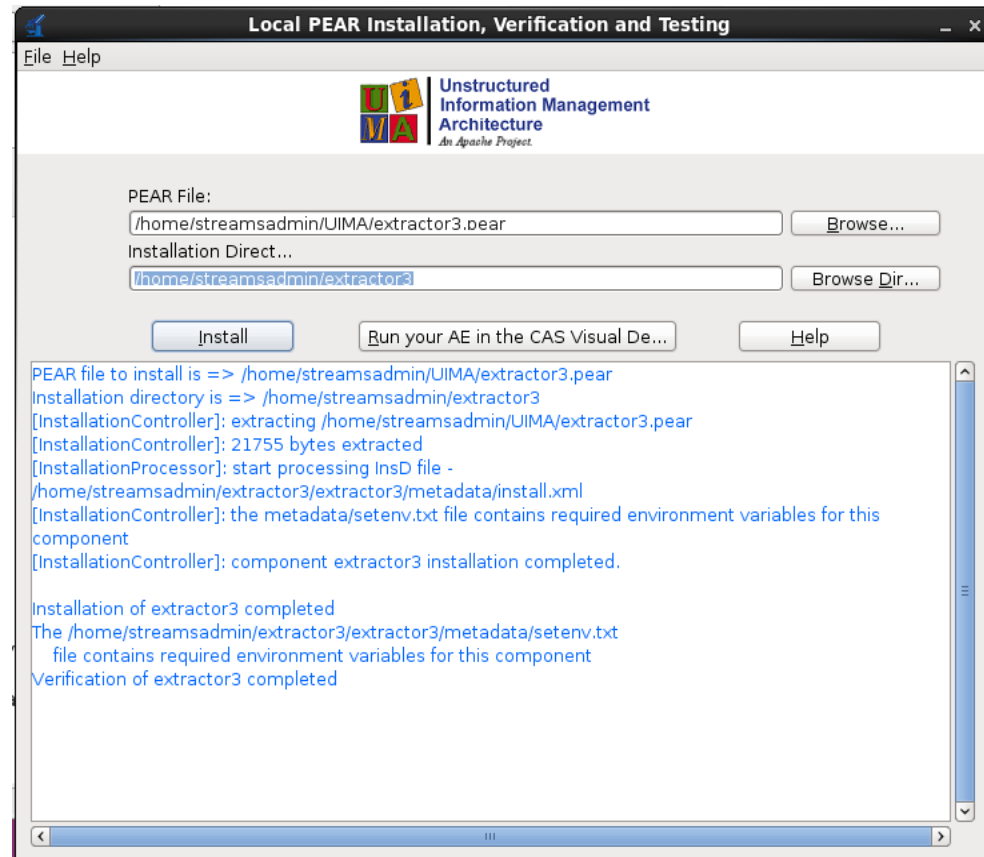
Run own UIMA project

- Eclipse -> Run -> Run Configurations...
 - Duplicate Java Application -> UIMA CAS Visual Debugger
 - In the Main tab:
 - Name: e.g. CVD – extractor3
 - Project: extractor3
 - Main class: org.apache.uima.tools.cvd.CVD
 - In the Classpath tab:
 - Add Projects...: extractor3
 - Advanced -> Add Folders -> extractor3 -> bin
 - Ensure that following external jars are added:
 - uima-core.jar, uima-tools.jar
 - Apply, Run (opens the CVD)
- In the CAS Visual Debugger (CVD):
 - Enter a text containing “charger broken” in the Text area
 - CVD -> Run -> Load AE -> select analysis engine descriptor xml from your project e.g. /home/streamsadmin/workspace_UIMA/extractor3/desc/aeDescriptor.xml
 - CVD -> Run -> Run AE Descriptor
 - In the Analysis Result top area you can see the number of created annotations in the brackets behind the Annotation Index (e.g. [3])
 - Selecting the Annotation index, opens the details below
 - Selecting one annotation below, marks the annotated area in the text



Create and test a PEAR file

- In the eclipse project context call Create PEAR File
 - e.g. Eclipse -> Project Explorer -> extractor3 -> context: Generate PEAR file
 - Component ID: extractor3
 - Component Descriptor: desc/aeDescriptor.xml
(Browse -> select: extractor3->desc->aeDescriptor.xml)
 - Next, Next: To pear file: e.g.
/home/streamsadmin/UIMA/extractor3.pear
 - Finish, ok
- Test the pear using the PEAR installer tool
 - From a shell call: runPearInstaller.sh
 - PEAR file: e.g.
/home/streamsadmin/UIMA/extractor3.pear
 - Install Directory: e.g. new dir
/home/streamsadmin/extractor3
 - Install



References

- Getting Started: Java UIMA Framework and SDK
<https://uima.apache.org/doc-uima-examples.html>
- UIMA Tutorial and Developers' Guides
https://uima.apache.org/d/uimaj-current/tutorials_and_users_guides.html#ugr.tug.aae.getting_started