

UIMA Tutorial and Developers' Guides

Written and maintained by the Apache UIMA Development Community

Version 2.3.0-incubating

© 2004, 2006

© 2006, 2010 ###

#####. ##### ## ## ##### #####
(###). ##### ## ## #####
#####, #####,

##. ##### ## ## #####
##, ## ##### ## ##
##.

#####. ### ## ##### ## ## #####
#####, ##### 2.0 (#####); ## ## ## ##
##. ## ## ## ##

[###://###.#####.###/#####/#####2.0](#)

##, #####
##, #####
##, #####. ## ##
#####.

#####. ### ##### ## ## ## ## ##
##. ## ## ## ##
##.

##, 2010

Table of Contents

1. ##### & ## ##### #####	1
1.1. #####	2
1.1.1. #####	3
1.1.2. ##### ##### #####	5
1.1.3. ##### #####	6
1.1.4. #####	9
1.1.5. #####	12
1.2. #####	15
1.2.1. #####	15
1.2.2. #####	19
1.3. #####	22
1.3.1. #####	22
1.3.2. #####	26
1.3.3. #####	27
1.4. #####	28
1.5. #####	29
1.5.1. #####	29
1.5.2. #####	31
1.5.3. #####	31
1.5.4. #####	34
1.5.5. #####	42
1.5.6. #####	44
1.5.7. #####	44
1.6. #####	45
1.7. #####	46
1.8. #####	47
1.8.1. #####	47
1.8.2. #####	48
1.8.3. #####	48
1.8.4. #####	48
1.8.5. ##### (#####)	49
2. ### #####	53
2.1. ###	54
2.2. ###	55
2.2.1. ###	55
2.2.2. ###	59
2.3. #####	60
2.3.1. #####	61
2.4. #####	61
2.4.1. #####	61
2.4.2. #####	68
2.4.3. #####	68
2.5. #####	71

2.5.1. #####	73
2.5.2. #####	74
2.5.3. #####	76
2.6. #####	77
3. #####	79
3.1. #####	79
3.2. #####	80
3.2.1. #####	80
3.2.2. #####	81
3.2.3. #####	82
3.2.4. #####	82
3.2.5. #####	83
3.2.6. ##### & #####	85
3.2.7. #####	86
3.3. #####	86
3.3.1. #####	87
3.3.2. #####	87
3.4. #####	89
3.5. #####	91
3.5.1. #####	91
3.5.2. #####	94
3.6. #####	96
3.6.1. #####	96
3.6.2. #####	98
3.6.3. #####	100
3.6.4. #####	101
3.6.5. ##### (###)	102
3.6.6. #####	105
3.7. #####	107
3.8. #####	108
3.9. #####	110
4. #####	113
4.1. #####	113
4.1.1. #####	113
4.1.2. #####	114
4.2. #####	117
4.3. #####	118
4.4. #####	119
4.5. #####	119
4.6. #####	120
5. #####, ##### & #####	123
5.1. #####	123
5.1.1. #####	123
5.1.2. #####	123
5.2. #####	123

5.3.	#####	###	#####	####	####	124
5.3.1.	#####	####	####			124
5.3.2.	#####	####	####			124
5.3.3.	#####	####	####	####	#	####	125
5.4.	###	####	#####	#####		125
5.5.	#####					126
5.5.1.	#####	#####	#####			126
5.5.2.	#####	####	##	#####	####	126
5.6.	#####					126
6.	#####	###	####			129
6.1.	###	####	###	####		129
6.1.1.	#####	###	####	###	####	129
6.1.2.	#####/#####	####	####	##	#####	130
6.2.	#####	#####				130
6.2.1.	#####:	#####				130
6.2.2.	#####:	#####	#####			131
6.2.3.	#####	###	#####			131
6.3.	####	#####	&	####	###	####	131
6.4.	####	####	#####			132
6.4.1.	####	#####	##	##	#####	#####	133
6.4.2.	####	#####	##	#	####	#####	134
6.4.3.	###	####	###	#####	#####	134
6.4.4.	####	#####	##	#	####	#####	135
6.4.5.	####	#####	###	#####	#####	136
6.5.	####	#####	###	#####	####	136
6.6.	#####	#####	#####			136
6.6.1.	#####	#####				136
6.6.2.	#####	#####				137
6.6.3.	#####	#####				137
6.6.4.	#####	###	#####	##	#####	138
6.7.	#####	###	#####			139
6.8.	####	#####	#####:	#1	###	#2	140
7.	###	#####				141
7.1.	#####	###	###	#####	####	141
7.1.1.	###	#####	#####	#####		141
7.1.2.	#####	##	#####	###	#####	142
7.1.3.	#####	####				143
7.2.	###	#####	#####			146
7.3.	#####	###	#####	##	#####	147
7.3.1.	#####:	#####	###	###	#####	147
7.3.2.	###	#####	###	####	#####	147
7.3.3.	#####	###	#####			149
7.4.	###	#####	##	####		149
7.5.	#####:	#####	###	#####		150
7.5.1.	#####	####				150

7.5.2. ### #####	151
7.6. #####	152
7.6.1. ### #####	152
7.6.2. #####	153
7.6.3. #####	155
8. ### & ###	157
8.1. #####	157
8.2. #####	157
8.3. #####	158
8.3.1. #####	159

[illegible]


```

#### descriptors/
tutorial/ex1/TutorialTypeSystem.xml.(####
## examples ####,####
#### 3.2,####
#####)
## uimaj-examples ####
#### descriptors/tutorial/ex1/TutorialTypeSystem.xml.#####
##### → #####.
####,####
###:

```



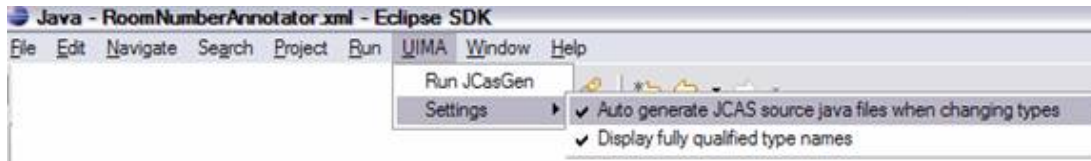
```
<?xml version="1.0" encoding="UTF-8" ?>
<typeSystemDescription xmlns="http://uima.apache.org/resourceSpecifier">
  <name>TutorialTypeSystem</name>
  <description>Type System Definition for the tutorial examples -
    as of Exercise 1</description>
  <vendor>Apache Software Foundation</vendor>
  <version>1.0</version>
  <types>
    <typeDescription>
      <name>org.apache.uima.tutorial.RoomNumber</name>
      <description></description>
      <supertypeName>uima.tcas.Annotation</supertypeName>
      <features>
        <featureDescription>
          <name>building</name>
          <description>Building containing this room</description>
          <rangeTypeName>uima.cas.String</rangeTypeName>
        </featureDescription>
      </features>
    </typeDescription>
  </types>
</typeSystemDescription>
```

1.1.2. Generating Java Source Files for CAS Types

#####, ## #####

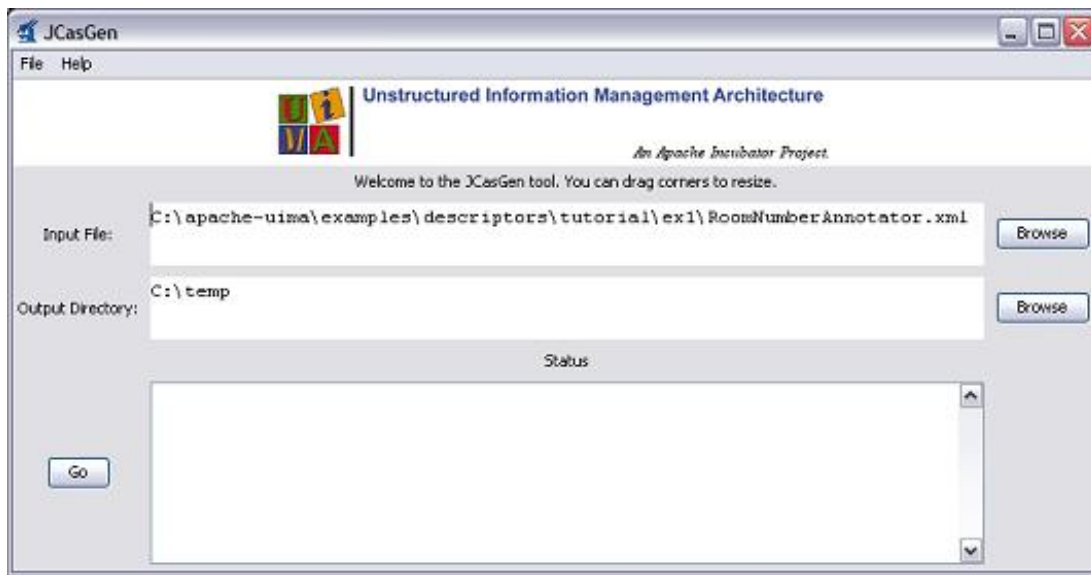
(######), #####. #####
(#####) ## ## ##, ## ##
##.

#####/##### ## ## ## ## (## ##
→ ###). ## ##### ## ## ##, #####
##:



```
### ##### ##### ### ##### #####.#####.#####.##### ##### ## #####
## src/org/apache/uima/tutorial/RoomNumber.java.### ##### ## ## ## #####
##### ##### ## ## #####.#####.
```

```
## ## ## ## ## ##### ## ###### ###### ###### ######,### #### ## ###### #####  
##### ## ##### ## ✂✂✂✂✂✂####.###### ##### # ##### #####  
##### ## ###### ## ###### ##### ##### ## ## ## ## ## ##  
##### ##.# ##### ######,### ### ##### ##### ##### ## ## /bin  
##### ## ## ##### ## ###### # ## ## ## ##  
#####:
```



```
##### (#####.###) #####
##### (### ### ##### ##### ##### ##### ##### #####).
##### #####. ## ##### ##### ##### ##### ## #####, ### #####
##### ## ##### ##### ##### #####.
```


7,

1.1.3. Developing Your Annotator Code

```
##### (#####),
#####.### #####:
```

- initialize,
- process, ###

- destroy.

```
initialize ## ##### ## ### ##### #### # ## ## ##### ## ##### ##  
### #####.process ## ##### #### # ## ## #####.destroy  
## # ##### ## ### ##### #### # # ## ## ## #####.##### # #  
##### ##### ## #### ##### #### ##### ##### # ##, #####  
#####_#####, ##### ## ##### ## ## ##### ##### ##  
## ##### #####.
```

_#####;### #####

#####. ##### ## ##### ## #####;### ##### 5,
[123].

#####_##### #####;### ##
#####,## ##### ## #####
#####. ² ##### ## ## ##### ##### #####
#####.## ## ##### ## ## ##,## ##### ##
#####.

#####, ##### ## #####, ### #####, 0#####. ## ##### ## ## #####. ³.

```
### ##### ##### ### ##### ##### ##### #####,### ##
#####.### ### ##### ### ##### ## ### uimaj-examples/src/org/apache/
uima/tutorial/ex1/RoomNumberAnnotator.java.
```

```
#####: ## #####, ## ### ##### #####, #### ##### ##
##### ## ### ##### uimaj-examples, ## ### ##### src, ## ### #####
org.apache.uima.tutorial.ex1.
## #####, #### ## #####.##### ## ### #####, ##### ##
#####.
```

```
package org.apache.uima.tutorial.ex1;

import java.util.regex.Matcher;
import java.util.regex.Pattern;

import org.apache.uima.analysis_component.JCasAnnotator_ImplBase;
import org.apache.uima.jcas.JCas;
import org.apache.uima.tutorial.RoomNumber;

/**
```

```

2 #####
##### JCas.class.##### process( AbstractCas cas ) #####,#####
cas ##### JCas.

```

```
3 ##### ## ##### ## ## ## ##### ## ##### ## ##, ## ## ##, ## # 0#####
##### ## #####, # ##### ## ##### ## ##### ## ##### ## ## ## # ##### 0#####
#####
```

```

* Example annotator that detects room numbers using
* Java 1.4 regular expressions.
*/
public class RoomNumberAnnotator extends JCasAnnotator_ImplBase {
    private Pattern mYorktownPattern =
        Pattern.compile("\\b[0-4]\\d-[0-2]\\d\\d\\b");

    private Pattern mHawthornePattern =
        Pattern.compile("\\b[G1-4][NS]-[A-Z]\\d\\d\\b");

    public void process(JCas aJCas) {
        // Discussed Later
    }
}

```

```

### ### ### ### ###,#####,#####
##### ### ### ## ### ## ### #####. ##### ### ### ###
##### ## ### ##### ## ### #####,### ## # ### ## ###
#####. ## ### ##### ## ##### ##### ## ### ##### 1.4. ##
## ### ##### ## ### ## ### ##### ## ### #####,### ## ### ##
##### ## ## ##### ## ### ##### ## ### #####.#####.

```

```

### ##### ## ## ## ##### ## ##### ## process. ##### ##
##### ## ## ## ##### ## ## #####. ##### ## ##
##### ## # #####; ##### ## ## ##### ## ## ##
#####.4

```

```

public void process(JCas aJCas) {
    // get document text
    String docText = aJCas.getDocumentText();
    // search for Yorktown room numbers
    Matcher matcher = mYorktownPattern.matcher(docText);
    int pos = 0;
    while (matcher.find(pos)) {
        // found one - create annotation
        RoomNumber annotation = new RoomNumber(aJCas);
        annotation.setBegin(matcher.start());
        annotation.setEnd(matcher.end());
        annotation.setBuilding("Yorktown");
        annotation.addToIndexes();
        pos = matcher.end();
    }
    // search for Hawthorne room numbers
    matcher = mHawthornePattern.matcher(docText);
    pos = 0;
    while (matcher.find(pos)) {

```

⁴##### 1 ## ##### ## #####,#####.#####
 ##### ## ## ##### ## ##### ## #####.### ## ## ## ##;#####
 ##### ## ## ##### ## ##### ##/#####() #####,##### ## ##
 # ##### (# #####) ##### ## #####

```
// found one - create annotation
RoomNumber annotation = new RoomNumber(aJCas);
annotation.setBegin(matcher.start());
annotation.setEnd(matcher.end());
annotation.setBuilding("Hawthorne");
annotation.addToIndexes();
pos = matcher.end();
}
}
```

```
### ##### ##### ## ##### ## ### #####.##### ##### ##### ## ## ##### ## ##### ## ##
##### ## ## ##### #####. ##### ## ##### ##, ##### ##### ## ## ##### ##
##### # ## ##### ##### ## ##### ##### ## ## #####:
```

```
RoomNumber annotation = new RoomNumber(aJCas);
annotation.setBegin(matcher.start());
annotation.setEnd(matcher.end());
annotation.setBuilding("Yorktown");
```

```
### RoomNumber ##### ## ##### ##### ## ## ##### ##### ##### ## ## #####
##### ##### ## ## ##### ## ##, ## ##### ## ## ##### #####.
```

```
#####, ## ##### annotation.addToIndexes() ## ## ## ## ##### ## ## #####
##### ## ## ##. ## #####, ## ## ##### ##### ## ## ##### ## ##
##### ##### ## ##### ## ## ##### ## ## ##### ##### ##### ## ## ## ##
#####. ##### ##### ## ##### ##### ## ## ##### ## ##### ##### ##
#####.
```

```
####: ## ## ##### ## ## ##### ## ## #####, ## ##### ## ##### ##
##### #####, ##### ## #####.
```

```
####: ## ## ##### ## ## addToIndexes() ## ##### ##### ## ## ##
##### ## uima.tcas.Annotation, ## ##### ## ## ## ## ##
##### ##. ## ## ## ## ##### # ##### ##, ## ## ##### ## ##
##### ##### ## ## ##: ## ##### 4, ~~~~~ ## ~~~~~
~~~~~ ## ##### 2.4.1.7, ##### ##### ## ~~~~~
~~~~~ ## #####.
```

```
##### ##### ## ## ## #####. ##### ## ## ## ## ##
#####.
```

1.1.4. Creating the XML Descriptor

```
### ##### ##### ##### ##### ##### ##### ## ##### ##
##### ## ## ## ## ##### ##### ## ## ##### ##### ## (#) ## ##
##### ##### ## ## ##. ##### ## ## ## ## ## ~~~~~
~~~~~. ## ##### #####:
```

- ####, #####, #####, ## #####

- #####, #####
#####
- #####

#####.

```
# #####  
##### descriptors/tutorial/ex1/RoomNumberAnnotator.xml. ## #####  
#####. ##### ## ##### ## ##### ## ##### ## ##### → #####  
#####.
```

```
###: ## #####,### ## ##### ##### ## ### ## ## ## ## ## #####
##### ##### ##### ##### ##### ## ##### ##### #####,### ##
##### ##### #####. ##### ##### ## ##### ## ##### ## ##### #####.
```

```
## ### ## ## ##### ,### #### ## #### ##### #####  
#####. ## ##### 1.8, ##### [47] ## ##  
## ## ##### ## ## ##. ## ## ##  
## ## ## ## ## ## ## ## ## ##  
#####.
```

#####; ## ### ### ###
#####. ### ##### ##### ## ##### ##### #####, ### ##### 1,
~~~~~  
~~~~~.

```
### ##### ## ## ##### ##### ##### ## ## ##### ##, ##### #####
## #####:
```

RoomNumberAnnotator.xml

Overview

▼ Implementation Details

Implementation Language ☐ C/C++ ☒ Java

Engine Type ☒ Primitive ☐ Aggregate

▼ Runtime Information

This section describes information about how to run this component

- ☒ updates the CAS
- ☒ multiple deployment allowed
- ☐ Outputs new CASes

Name of the Java class file

▼ Overall Identification Information

This section specifies the basic identification information for this descriptor

Name

Version

Vendor

Description:

Overview | Aggregate | Parameters | Parameter Settings | Type System | Capabilities | Indexes | Resources | Source


```
##### ##### ## ##### ## ### ##### ##### ##### ##### (##). ### #####
##### ## ##### ##### ##### ##### ##### ##### ## ### # ##### ## (##### ##
##### ## # ##### #####), ### ##### ##### ##### ## ##### ## #####. #####,
## ##### ## ##### ##### ##### ##### ##### (### ##### ## #####
## ### ##### #####). #####, ## ### ##### ## ### ##### ## #####
##### ##### ## ### #####.
```

```
### ##### ## ##### ##### ## ## ##### ## ##### ##### ##### ## ##
##### #####. ### ## ##### ## ##### ##### ##### ## ##### ## ##
##### #####. ## ### #####, ##### ## ##### ##### ## ## ##.
```

```
### ##### ##### ## ##### ## ##### ##### ##### ## ##### ## ## #####
##### 1.1.1, ##### [3]. ## ##### ##, ## ## ##### ##### ## ##
##### ##### ## ##### ##### #####, ##### ## ##### ##### #####
##### ##, ## ##### ##:
```

Type System Definition

Types (or Classes)

The following types (classes) are defined in this analysis engine descriptor. The grayed out items are imported or merged from other descriptors, and cannot be edited here. (To edit them, edit their source files).

| Type Name or Feature Name | SuperType or Range | |
|---|----------------------|--|
| <input type="checkbox"/> org.apache.uma.tutorial.RoomNumber | uma.tc.as.Annotation | <input type="button" value="Add Type"/> |
| building | uma.cas.String | <input type="button" value="Add..."/> |
| | | <input type="button" value="Edit..."/> |
| | | <input type="button" value="Remove"/> |
| | | <input type="button" value="Export..."/> |
| | | <input type="button" value="JCasGen"/> |

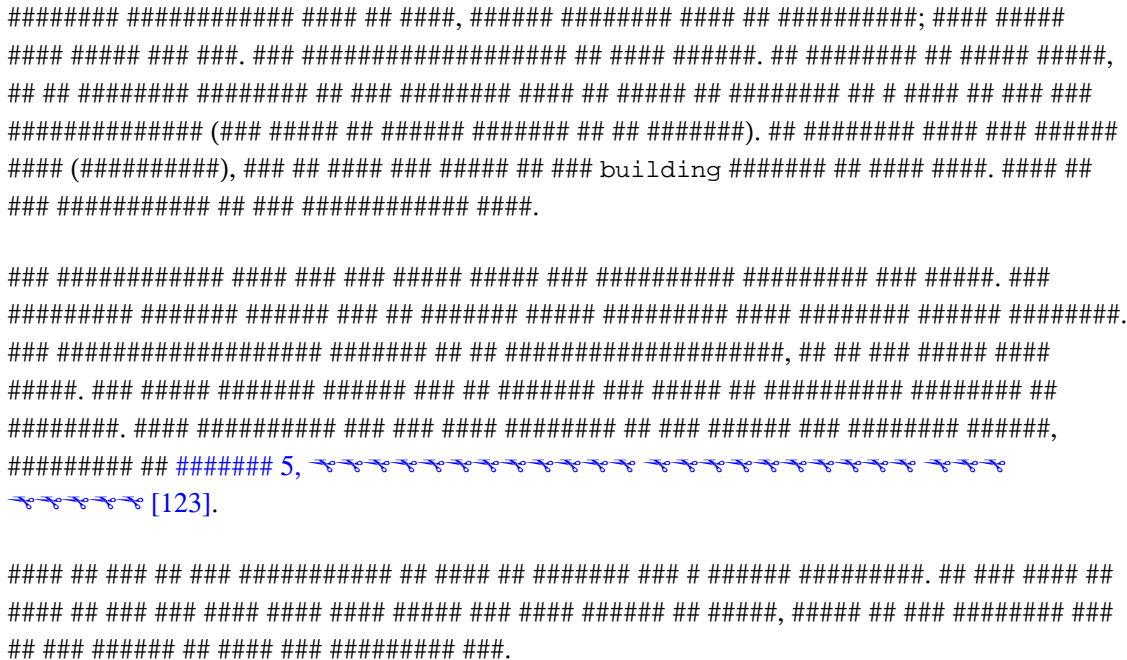
< ||| >

Imported Type Systems

The following type systems are included as part of this one.

| Kind | Location/Name |
|-------------|------------------------|
| By Location | TutorialTypeSystem.xml |

```
## ### ##### ##, ## ##### ## ##### ##### ##### ## #####, ## ##### ## ##
##### ## ## ##### #####. ### ##### ##### ## ##### #####:
```

[illegible]

```
#####, ##### ## ##### ##### ##### ##### ##### ##### #####
#####. (## ## ####, ##### ## ## ##### ## ## → ## ... → ## ##### ##### #####
## ## ##### ##, ##### ## ##### ##### #####.)
```

```
### ##### ## # ##### ##### ##### #####:
```

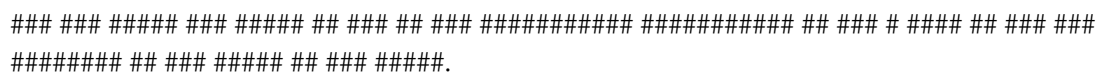
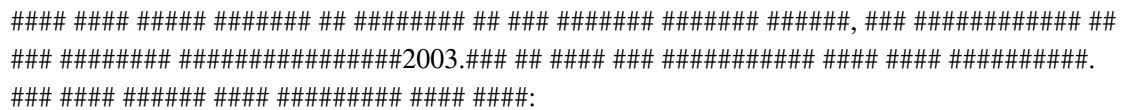


```
##### ## ## ##### ## #####:
```

1. #####
2. #####
3. ## ## ##### ## ## ##### (##) ## ## ##
4. (#####) ## ## ##, ##### ## #####, ##### ##### ## ##
#####. ## #####, ## ##### ##### ##### ## ## ##
<####>...</####> #####.
5. ##### ## ##
6. #####

```
### ## ##### ##### ## ## ##### ## ## ##### ## ## #####
##### ## ## ##### ##### ##### ##### # <where-you-installed-uima-
e.g.UIMA_HOME> /examples/descriptors/tutorial/ex1/RoomNumberAnnotator.xml
. ## ## ##### ## ## ##### ##### ## ## ##### (##### ## ##
##### ## ## ## ## ##### ##### ##### #####). #####
## ## ##### ## #####
```

```
#### #####, ## ##### ##### #####
```



```
####: ### ##### ##### ##### ##### ##### ##### ##### ## ##### ## #####
## ## ##, ## ## ##### ## ##### ## ## ##### ##### ## ##
##### (## ##### 1.1.4, ##### ## ## ##### [9].
```

```
## ## ## ## ##### ## ## ## ## ##### # ## ## ## ## ## ##
##### ## ## ## ## ## ##.
```

1.2. Configuration and Logging

1.2.1. Configuration Parameters

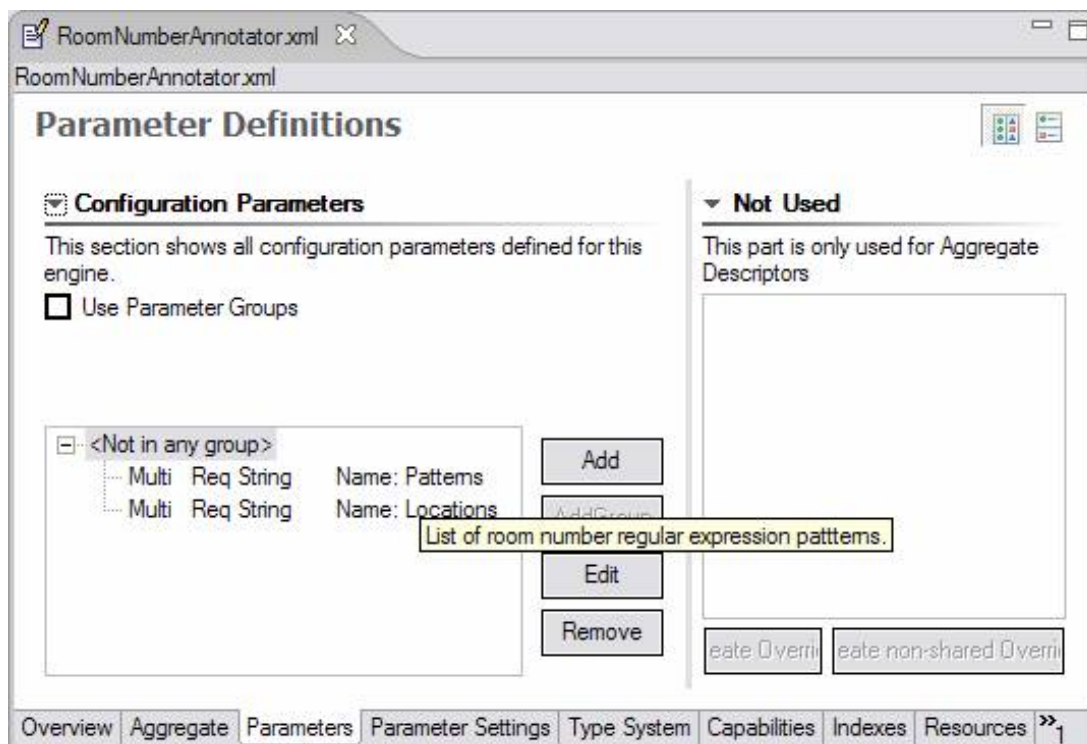
```
## ##### ##### ##### ## ## ##### ##### ##### #####
##### ## ##### ## ##, ##### ## ##### ## ## #####. ## #####,
## ##### ## ## ## ## ##### ## ## ##### ## ## #####
#####, ##### ## ## ##### ## ## ## ##### ##### ## ## #####
#####. ##### ## ## # ## ##### ##### ##### ## # ## ## ##, # #####
##### ## ## ## ##### #####.
```

```
#### ##### ##### ## ##### ##### ##### ## #####.
## ##### ## ## ##### ##### ## ## #####, ##### ## ##
##### ## #####.
```

1.2.1.1. Declaring Parameters in the Descriptor

```
## ##### ##### descriptors/tutorial/ex2/RoomNumberAnnotator.xml ##
## ## ## ## ##### ## ## ##### ##### ##### ## ##
##### ## ## ## ##### ## ##### ##### ## ## #####
#####.
```

```
#####, ## #####, ## ## ##### ##### ##### ## ## #####
#####, ## ## ## ## ## ##### ## ## (##### ## ## ##### ##
## ## ##), ##### ## ##### ## ##:
```



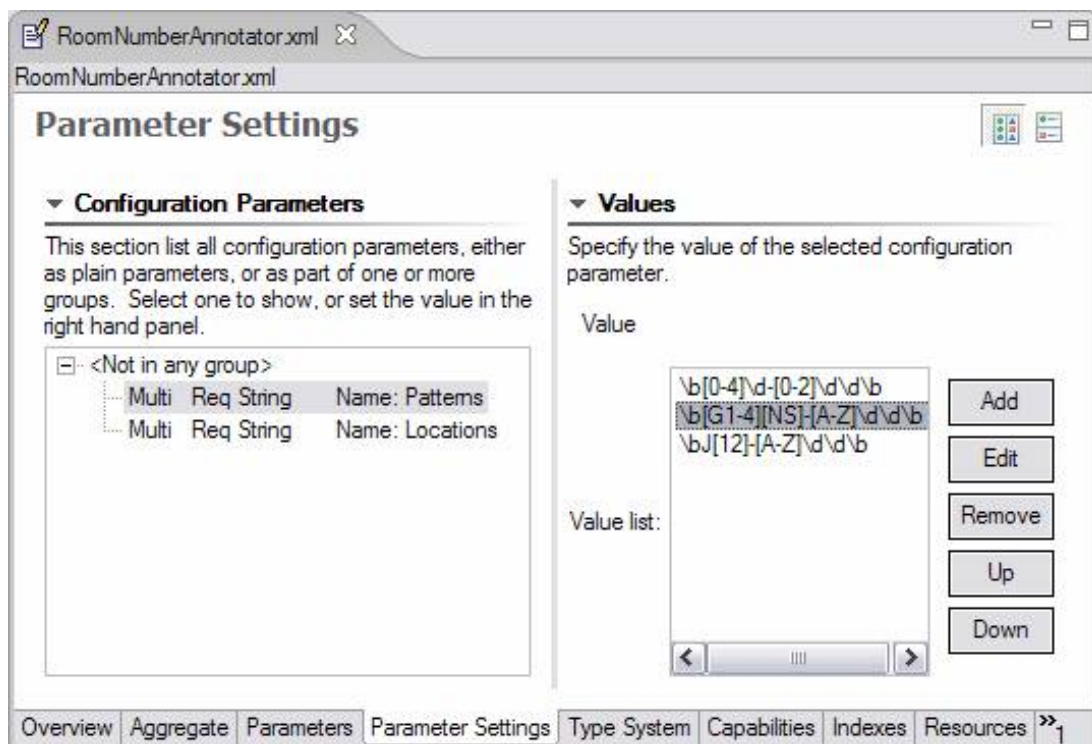
```
### ##### # ##### ## ##### #####. ## ##### ##, ###
##### (## #####) ## ##### ## ##### ## ##### ##
#####. ##### ##### ## ##### ##### ##### ##:

```

- ##### # ## ##### ## ##### ## ##### ## ## #####
- ##### # # ##### ##### ##### ## ## ##### ## ## #####
- ##### # ## ##### ## ## ##### ##### # ##### ## ## ##, #####,
 #####, ## #####.
- ##### # ##### ## ## ##### ## ##### ##### (## #####), ##### ##
 ##### ##### ## ##### #####. ##### ##### ## Multi.
- ##### # ##### ## # ##### ## ##### ## ##### ## #####. ##### ##### ## Req
 (## #####).

```
#### ## ## ##### ## ##### ## ##### ## ##### ## ##### ##.
```

```
####, ##### ##### ## ##### ## ## ##### ## ## ##### ##:
```



```
#####  
#####  
#####  
#####  
#####
```

1.2.1.2. Accessing Parameter Values from the Annotator Code

```
### ##### org.apache.uima.tutorial.ex2.RoomNumberAnnotator ### #####
### #####. ### ##### ## ##### ## ### #####
### #####, ## ## # ##### ## #####
#####. ### ##### ##### ##### #####, ##
### ##### ##. ## ### ##### ## #####, ##### ## ### #####, ### #####
org.apache.uima.tutorial.ex2.#### ## ### ##### ##:
```

```
/**
 * @see AnalysisComponent#initialize(UimaContext)
 */
public void initialize(UimaContext aContext)
    throws ResourceInitializationException {
    super.initialize(aContext);

    // Get config. parameter values
    String[] patternStrings =
        (String[]) aContext.getConfigParameterValue("Patterns");
    mLocations =
        (String[]) aContext.getConfigParameterValue("Locations");
}
```

```
// compile regular expressions
mPatterns = new Pattern[patternStrings.length];
for (int i = 0; i < patternStrings.length; i++) {
    mPatterns[i] = Pattern.compile(patternStrings[i]);
}
}
```

```
#####  
#####  
#####  
#####  
#####  
#####
```

getConfigParameterValue #####
#####; #####

#####.
#####[] #####.

#####.

examples/descriptors/tutorial/ex2/
RoomNumberAnnotator.xml .
WatsonConferenceRooms.txt,

#1 #####.

```
## ## # ## ## ## ##### (###### ## ## ###/###5 #####) ##
org.apache.uima.analysis_component.AnaysisComponent (#### ## #####
##### #####_#####), ## ## ## ## ## ##
#####() #####, ##### ## ##### ## ## ##### ##### ##### ## ##
#####, ## ## ##### ##### ## #####.
```


#####

5 ####/#####.####


```
#####; ### ##### ## ## ## ## ## ##### ## ## ##### ##### ## #####
##### ## ##### ## ## ## ##### ##### ##### ## ## #####
#####.

### ##### ## ##### ##### ## ## ##### ## #####
##### # ## ##### 2, ~~~~~
~~~~~ ## ~~~~~ ## #####. ##### ##
##### ##### ##### ## ##### ##### ## ##### ##### ##
UimaContext.getConfigParameterValue, ##### ##### ## ## ## ## ## ## ##
##### ## ## #####.
```

1.2.2. Logging

```
### ## ## ##### # ##### #####, ##### ## ## ##### ## ##
####.####.#####.##### ##### ## ##### ## ## 1.4.

## ## ## #####, ##### ##### ## ##### ## ##. ## #####,
#### ## ## ## ## ##### ##### ## ## ## ##### ##### ## ##
####. ## ## ## ## # ##### ## ## ##### ##### ## ## ##
## ##### ## ##### ##, ## ## ## ## ## ##.

### ##### ##### ##### ##### ##### ## UimaContext #####. ## ##
##### # ##### ##### ## getContext().getLogger() ##### ## #####, ##
##### ## ## ## ## ## ## ## ## ## ## ## #####.

#### ## ## ##### ## ## ##### ##### ##
org.apache.uima.tutorial.ex2.RoomNumberAnnotator:

getContext().getLogger().log(Level.FINEST, "Found: " + annotation);

### ##### ## ## ## ## ## ## ## ## ## ##. ####, # ##### ##
##### ##### ## ## ## # ##### #####. ##### ##### ##
#####, ## ## ##### ## ## ##### ## ## ## ## ## ## ## ##
####, ## ##### ## ##### #####. ##### #####, ##### ##
##### #####, ## #####, ##, ##, ##, ##, ##, ##.

## ## ##### ##### ## ## ##### (## ## ##), ## ## #####
##### ##### ## ##, ##### ##### ## ## ## ## ##
#####, ## ##### ## ## ##.

## ## ##### ## ##### ## ## ## Logger.properties, ## ##### ## ##
## # ## ## ##.##, ## ## ##### ##### ##### (##### ##)
##### ## ## ##### ## ##, ## #####).
```

```
####: #### #####, ## ##.## ##, ## ##### ## ##
##### ## ## #####, ## #####, ## ## ## ##
##### ##### ## ## ## ## ##### ## #####
##### ## ## ##, ## #####. ##### ##### ##
```

1.2.2.1. Specifying the Logging Configuration

```
##### ##### ## ## ##### ## # ## ##, ## ## ## ## # #####. ##### ##
##### ##### ## # ##### ##### ##### ## ##### ##### ##### ## ##
#####. ## #####: .level= INFO

## ##### ##### ## ##### ##### ## ## ##, ## ##.

##### ##### ##### ##### ##### ##### ## ## ##, ## ##. ##
#####, ## ## ## ##.###.### ##### ## ## ## ##### #: com.xyz.foo.level
= SEVERE

## ## ## # ##### ##### ## ## ##### org.apache.uima.SampleAnnotator ##
## ## ## ## ##### ## ##### #: org.apache.uima.SampleAnnotator.level = ALL

##### ## ##### #####; ## # #####, ##### ## ## ##### ##
## Logger.properties ##### ## ## ## ##### ## ##### ## ## 1.4.
```

1.2.2.3. Format of logging output

```
## ##### ##### ## ##### ## ##### ## ## ##### ##### ##
#####, #####. ## ##### ##### ##### #####
## ##### ##### ## #####:

Timestamp - threadID: sourceInfo: Message level: message

##### ## #####:

7/12/04 2:15:35 PM - 10: org.apache.uima.util.TestClass.main(62): INFO:
You are not logged in!
```

1.2.2.4. Meaning of the logging severity levels

```
##### ##### ## ## ##### #####, ##### ## #####
##### ## ## ## 1.4 #####. ## ##### ## ##### ##
#####.#####.#####, ## ##### ##### ##### #:



- ## # ## ##### ##### ## ## ## ## ##.
- ## ##### ##### ## ##### ## ##.
- ##### # # ##### ##### ## #####. ##### #####
  ##### (#####) ## ##### ## initialize().
- ##### # # ##### ##### ## #####, ## #####, ##### ##
  ##### #: 192.168.120.12
- ##### # # ##### ##### # #####.
- ##### # # ##### ##### # #####.



#####, ##### #:

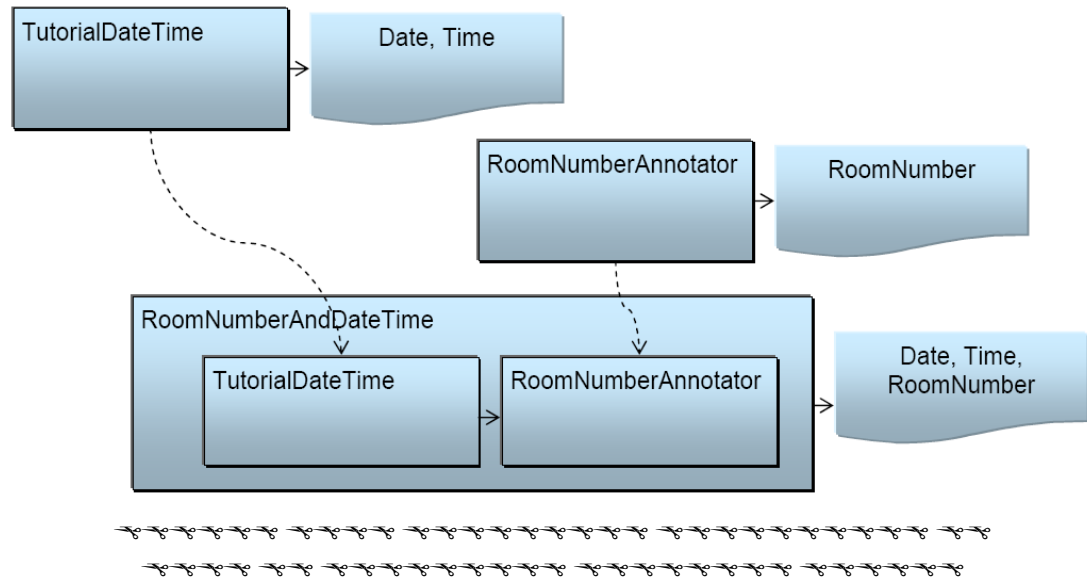


- ##### # # ##### ##### #####, ##### ## # #####
  ##### (#####).
- ##### ##### # ##### #####, ##### ## # #####
  (#####).

```

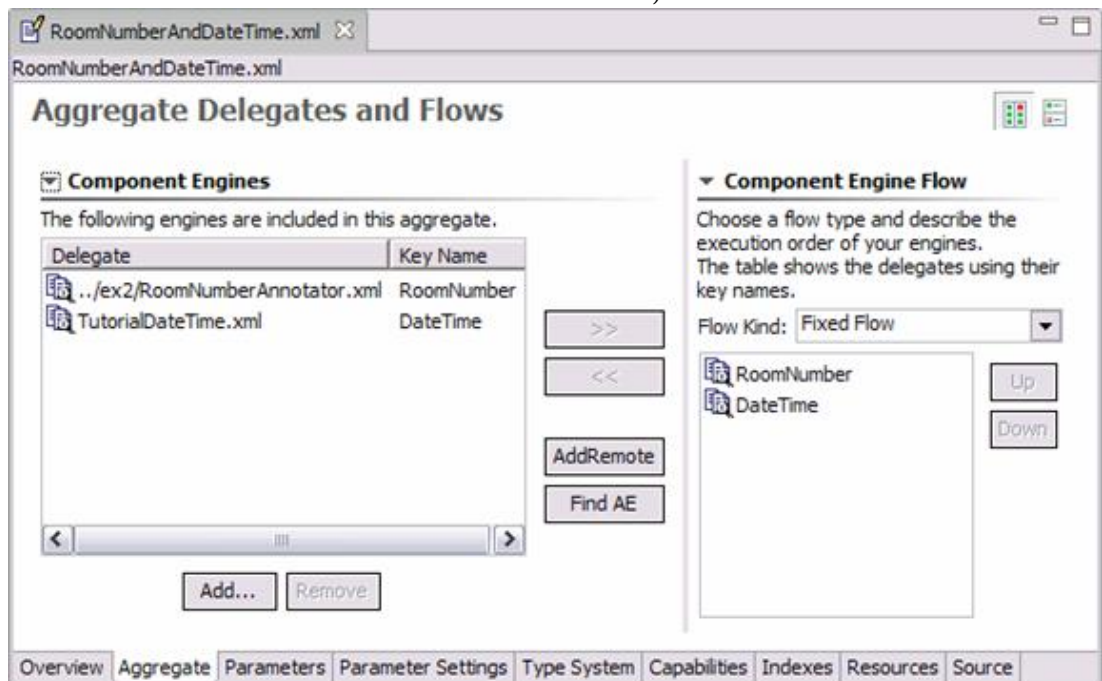
- ```
examples/descriptors/tutorial/ex3 ##### (# #####, #### # ##
descriptors/tutorial/ex3 #####), ## ##
#####. #### ##### ## ##
(### ##### #). ## ## ## ## ## ## ## ##,
#####. ## ## ## ## ## ## ## ##, ## ## ## ##
#####. ## ## ## ## ## ## ## ##.
```

```
#####
#####. ##### ## ##### ## ##### #####:
```

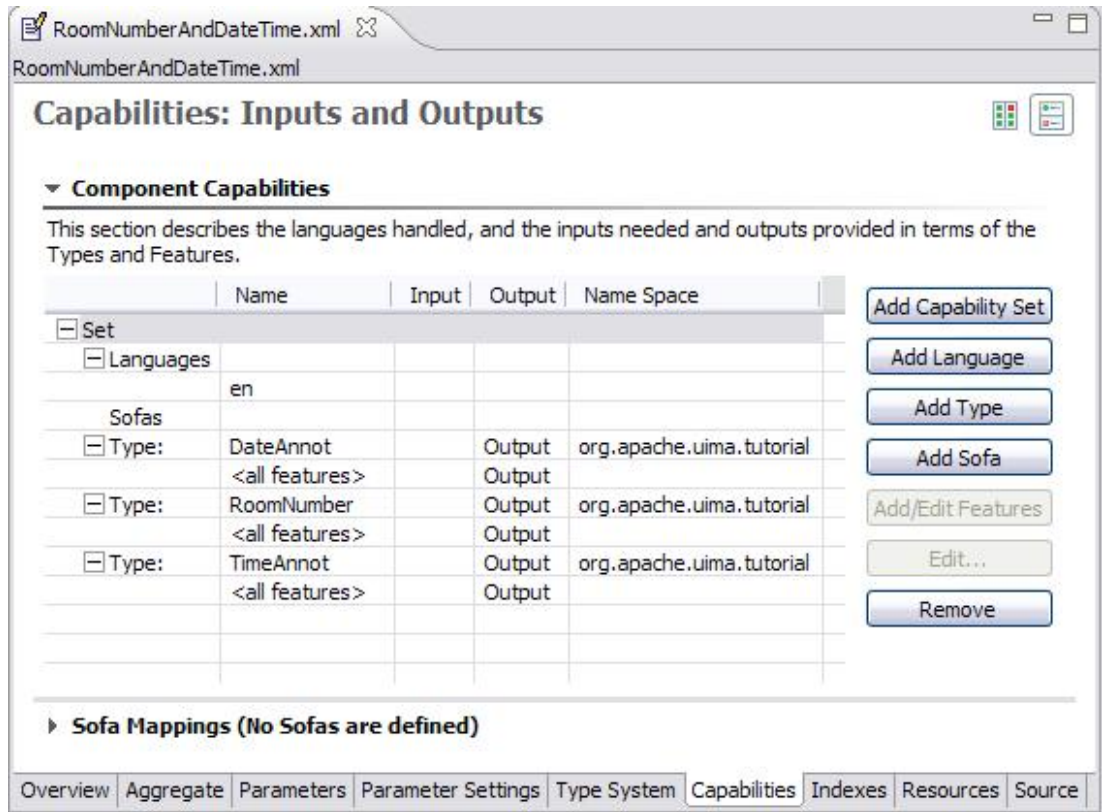


```
RoomNumberAndDateTime.xml, ##### ##
#####. ##### ## ## #####
descriptors/tutorial/ex3.
```

```
#####
#####. # ##### ##### ## ##### #####. (## ##### ##
#####., ## ##### 1.8, ##### ##### ## ##### [47] ### ##### ##
#####. #####. #####.)
```

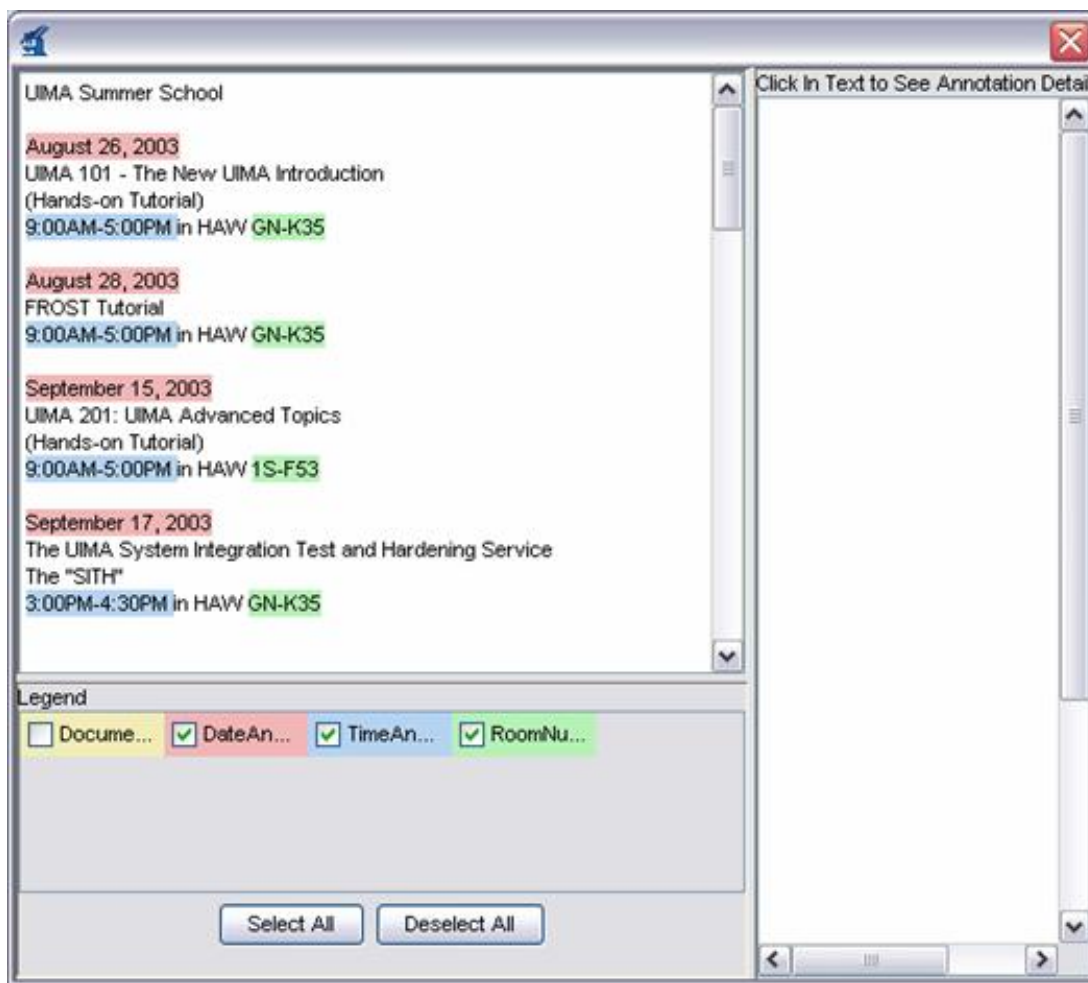








```
#####
(##) ##### ##### ## ##### ## ## ##### ##. ## ##### ##, ## #####,
##
#####.
```

```
##
##. ##### ## ## examples/descriptors/tutorial/ex3/
RoomNumberAndDateTime.xml ##### ## ## ## ## ##. ## ## ## ## ##
#####, #####, ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##:
```



### 1.3.2. AAEs can also contain CAS Consumers

```
#####, ##### ## ## #####
(## ##### 2, 
 [53], ## ## # ##### ## ##### #####
#####. ## ## ##### ## ## ##### ## ##
#####, ## examples/descriptors/
MixedAggregate.xml.
```

[illegible]



### 1.3.3. Reading the Results of Previous Annotators

```
###, ## ##### ##### ##### ## ##### ##### ##### ## ##### #####.
#####, ##### ##### ##### ## ##### #####. ## ##### ##### ##
###, # #####, ## ##
#####.
```

```
~~~~~ ## #####, ## ##### ## ##### ## #####
#####.
##:
```

```
FSIndex timeIndex = aJCas.getAnnotationIndex(TimeAnnot.type);
Iterator timeIter = timeIndex.iterator();
while (timeIter.hasNext()) {
 TimeAnnot time = (TimeAnnot)timeIter.next();

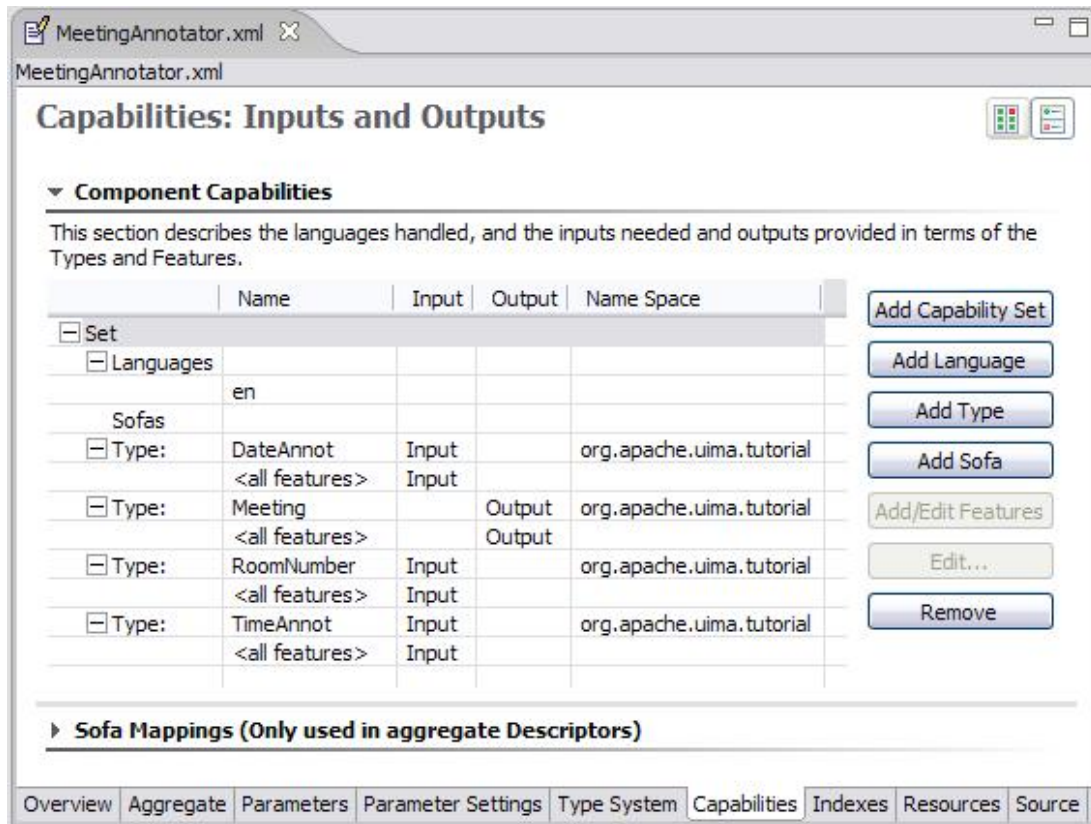
 //do something
}
```

```
####: ## ## ##### ## ## #####
JCAS.getJFSIndexRepository().getAllIndexedFS(YourClass.type), #####
YourClass ## ## #####.
#####
#####.
```

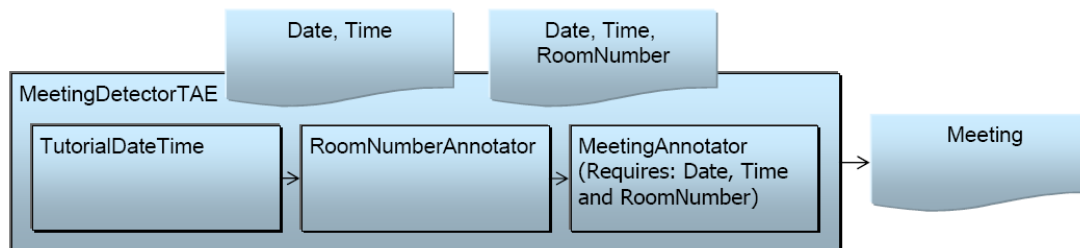
```
#####, ##### ## ## ## ##### #####
org.apache.uima.tutorial.ex4.MeetingAnnotator. ##### ##### ## #
#####, # ###, ## ## #####, ##### ## #####.
(##### ## ## ##### ##### ## ## ##, ## ## ## ##### ##### ##)
#####.)
```

```
#####, ## ##### ## ## ##
##, ## ## ## ## ## ## ## ## ## ## ## ## ##,
#####. ##### ## ## ## ## ## ## ## ## ## ##
##. ## ## ## ## ##, ## ## ##
##. ##### ## ## ## ##!
```

```
#####, ##### ## examples/descriptors/tutorial/ex4/
MeetingAnnotator.xml, ## ## ## #####. ## ##### #####
##
#####. ##### ## ## ## ## ## ## ## ## ##
#####:
```



```
##, ## ##### #####
##. ## #####
#####. ##, ## ##### ##
#####, ##### ## ## #####.
1.2, ## ##### ##### #####
[28]. ## ##### ## ##
examples/descriptors/tutorial/ex4/MeetingDetectorAE.xml . ##### ## ##
#####.
```



## 1.4. Other examples

```
##,
```

```
#####, ### ## ####,
##.
```

---

UIMA Version 2.3.0

Annotator & AE Developer's Guide

29

| ##### | #### ##### ## #####                                                                                                                                                                                                                                                                                                    | #####                                                                                                                  |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
|       | #### ## ##### #####<br>### ##### ## #####, #####<br>##### #####<br>## ##### #.                                                                                                                                                                                                                                         |                                                                                                                        |
| ##### | ##### ## ## ## ##. #####<br>## ## ##### ## ##<br>#####<br>##### (#); ###<br>##### ##<br>### #####, #####<br>## ## ##### ##<br>##### ## ## ##<br>## ## #####. ##<br>##### ##, ##<br>### #####.<br>## ## ##### ##<br>#####<br>##### ## ## ##<br>(##), ##<br>#####<br>##### ## ##<br>##### ## ##<br>##### ## ##<br>#####. | ##### ## ##, ##### ##/<br>##### ##                                                                                     |
| ##### | #### ##### ## ##<br>## #####, ## ##<br>#####<br>#####<br>#####. ## ##<br>## #####<br>#####,<br>##### initialize ##, ##<br>##### (#<br>#####) ## ##<br>#####. ##<br>#####<br>## ## ## ##<br>##### ## ## ##. ## ##<br>## ## ##### ##<br># ##/#####<br>#####<br>#####                                                     | ## #####<br>### #####, #####,<br>#####<br>##., ## ##<br>#####<br>##### ##<br>#####. ##<br>#####,<br>## #####<br>#####. |

| ##### | #### ##### ## #####                                                                                                                                                                   | #####                                                                                                                                                                  |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | #####<br>#####. ## destroy #### ##<br>## ##### ## #####<br>### ##### #####.                                                                                                           |                                                                                                                                                                        |
| ##### | #### ##### ## #####<br>## ## #####, ##### ##<br>##### ## ## ##<br>##### # ## ##### ##<br>### ##### ## ##<br>### ##### ## ##<br>#####.<br><br>### ##### ## ## #####<br>#####<br>#####. | # #####<br>#### #####, #####<br>## #####. #### ## ## ##<br>#####<br>## ##### ## ## ##. #####<br>#####<br>##### ## #####<br>##### ## ## #####<br>#####, #####<br>#####. |

## 1.5.2. Reporting errors from Annotators

```
#####: #####
(###
#####), #####), ##### ## ##### #####.
```

```
#####, ## ## ##### ##### ##### ##
#####, ## ##### ## ## ## ##, ##### ## ## ##
##, ## #####
#####. ##, ## ## ## ## (### ##) # ## ## ##, ##
#####.
```

```
####: ## ##### ## ## ## ## ## #####,
#####.
```

## 1.5.3. Throwing Exceptions from Annotators

```

#####. ##### ## ## ##### ## ##
##, ## #####
##:
```

```
throw new ResourceInitializationException(
 "The regular expression " + x + " is not valid.");
```

```
####, #####, ##### ## ## ## ##. ##
~~~~~  
#####. #### ## ##### ## #####
```

```
try {
    mPatterns[i] = Pattern.compile(patternStrings[i]);
}
```

```
catch (PatternSyntaxException e) {
    throw new ResourceInitializationException(
        MESSAGE_DIGEST, "regex_syntax_error",
        new Object[]{patternStrings[i]}, e);
}
```

```
##### ## #####_##### ##### ## ## #####
```

```
"org.apache.uima.tutorial.ex5.RoomNumberAnnotator_Messages".
```

```
##### ##### ## ##### ##### # ##### ##, ##### ##, ##### ##, #####. #####,
##### ##.##### #####, ##### # ##### # ## ##### ##. ## #####, ##
##### ## ##### ## ## #####, ## ## ##### ##.#####.#####.##5, #####
##### #####_#####.#####.##### # #####, ## ## ##
##### ## ## uimaj-examples.jar ##### ## ## ## org/apache/uima/tutorial/ex5/
RoomNumberAnnotator_Messages.properties. ## ## ## ## ## ## ## ##
### ##:
```

```
regex_syntax_error = {0} is not a valid regular expression.
```

```
##### ## ## ##### ##### ## ## ##### ##### ## #####. ## #####
{0} ##### ## ##### ## ## #####() ##### ## ## ##### ##### ## ## #####
##### # ## ## ##, ## ##### ##### ##### #####. ##
##### ##### #####, ##### ##### ## ## ##### ##### ##
{1}, {2}, ## ## #.
```

```
## # ##### ##### ## ## ##### ## ## ## ## ##### #####,
### ##### ## UIMAException.STANDARD_MESSAGE_CATALOG (##### ##
#org.apache.uima.UIMAException_Messages # ## ## ##### ##### ## ##
##### ##### ## ##### ## ## uima-core.jar ##### ## org/apache/uima/
UIMAException_messages.properties # ## ## ## # ## ## ## ## ##
##### ##### ## ##### ## ##.
```

```
## ## ## ## #####_#####_##### #####, ##### ## ## ##### ##### ## ##
examples/descriptors/tutorial/ex5/RoomNumberAnnotator.xml, ##### ##### ##
##### ## ##### ##### ##### ## ## ##### ##### #####.
```

```
## #####, ##### ## ## ##### ## ## ## ## ## ## ## #####
#####:
```

```
##### # ##### ## ##.##### #####, ##### ## ##### ##### ## ##
##### #####, ##### ## ## ##### ## ## ## ## ##_#####_#####
#####. ## ##### ## ## ## ## ## ##### ## ## ##### ##
### ###6 ##### ## ## ##.#####.#####.
```

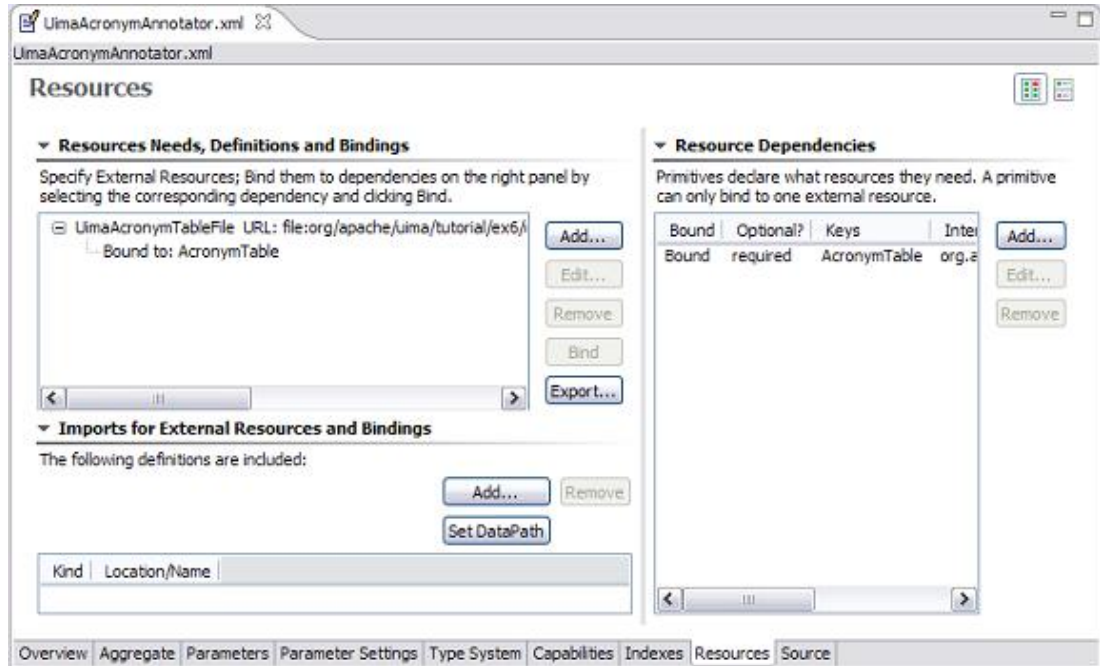
```
### ##### ##### ##### ## ## ## ## ## ## (## ## ## ## ##
#####. ## ##).
```

<sup>6</sup> #####://#####.##.##/#2##/1.5.0/#####/#####/#####.#####(#####.##.#####)

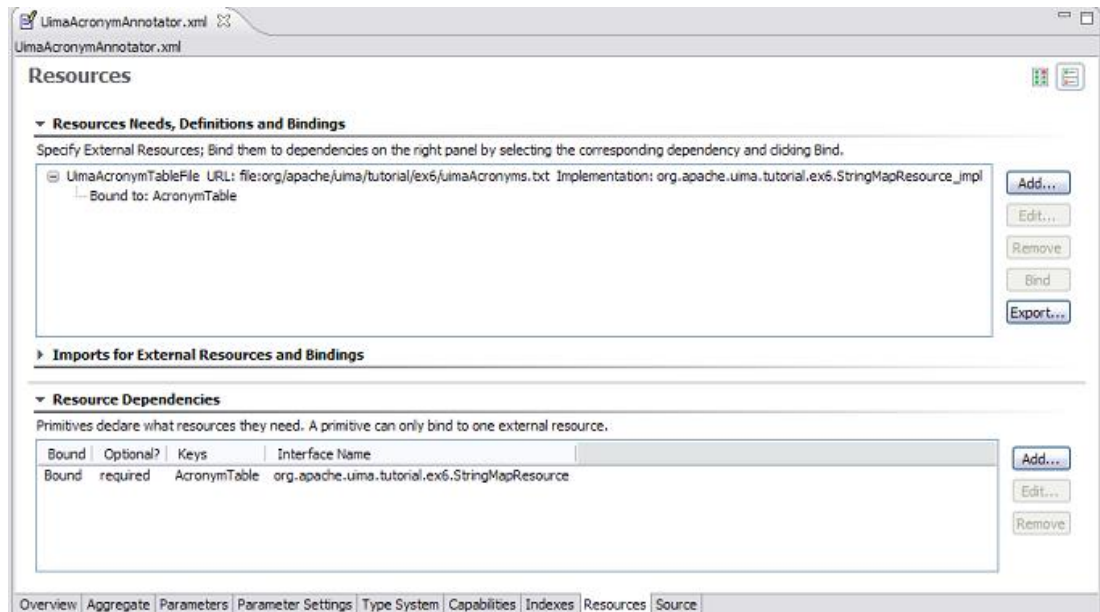
8 #####://#####.###.###/#2##/1.5.0/#####/#####/#####.####



```
#####, ##### ## ### examples/descriptors/tutorial/ex6/UimaAcronymAnnotator.xml
#####.
```



```
### ##### ## ### ##### ## ### ##### ##### ##### #####. ### ###
##### ## ### ##### ## ### ##### ## ##### ##### #####
#####, ## # ##### #####. ### ### ##### ##### ##### ## ###
##### ## ##### ##### ## ##### ## ##### #####, ##### ##
### #####. ##### ## ##### ##### ##### #####:
```



```
### ### ##### ## # ##### ## ##### ## ## ## ## ## ##.
```



### 1.5.4.3. Declaring Resources and Bindings

```
### ##### ##### ## #### ## ##### ## #####,###
UimaAcronymTableFile.#### ####,### ##### ##### ##### ## ##
#### ####.#### #### ## ## ##### ## (#.#.### #### ##### #### ####:/ ## ####://,##
####://##.####.###/),### #### ## ## ##### ##### ##### ## ##### ##### ## ####
##### #### #####,## ##### #####.##### ## # ##### ###,##### ##
##### ## ##### ### ##### (###/## #####),## #### ## #### #####.## #### ####,
### #### org/apache/uima/tutorial/ex6/uimaAcronyms.txt ## ##### ## uimaj-
examples.jar,##### ## ## #### #####.## #### ## #### ##### ## ##
##### ## ##### ##### #####.
```

```

### ##### ##### ## ### ##### ## #####, #####
### ### AcronymTable, ##### ## ### #####, ##
### ##### ##### UimaAcronymTableFile. ##### ## ##### ## #####;
### ##### ## ##### ## ### ##### UimaMeetingDetectorAE.xml #####. ##### ##
##### ##### ## ##### ##; ## ## ## ## ## ## #####
##### ## # ##### ## ## #####.

## ### #####, ##### ## #####
#####. ## ##### # #####, ## ##### ## ##### (#####
#####), ## ## #####, ## ##### ## Bind
#####, ##### ## ##### ## ## ##### ## #####.

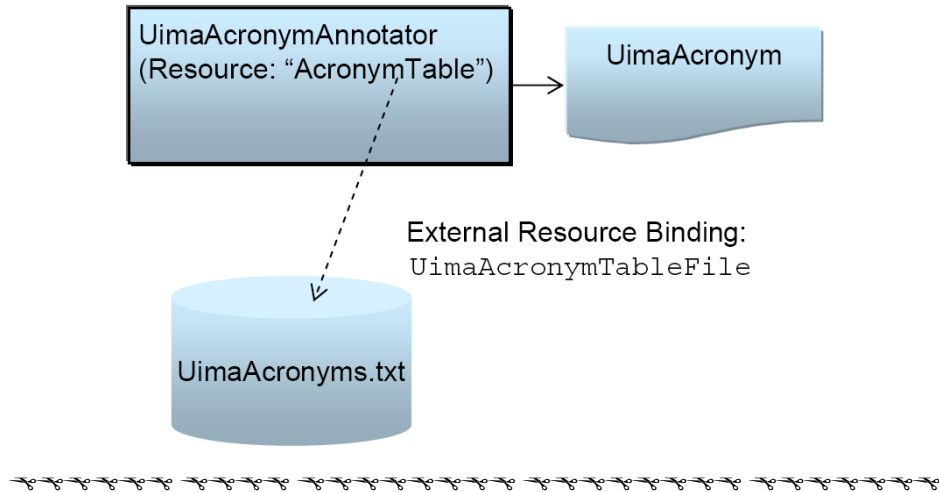
#### ## #####, ## ##### # ##### ##
StringMapResource_impl ### ##### ## ##### ## #####. ####
##### ## ##### load #####, ##### ## ##
#####, ##### ## ## ## ## ## ## ##
#### ## ##### ## ## #####; ## ##### ## ##
####, ## ## ## load ##### ## ##### ## ##. ##
## ##### ## ##### ## ##. ##
#### ##### ## StringMapResource #####. #####
##### ## ##### ## ## ## ##. ##### ## ##,
## ##### ## ## ## ##, ## ## ## ##
##### ## #####.

#### ## ## ##### (#.#. ##### ## ##
#####) ##### ## ##### ## ## ##, ## ##
#####, 0#####, ## ## ## ## ##.
(##### ## ## ## ## ## ##, ## ##, ##
# 0##### ## ## ##, # ## ## ## ## ##
##### ## ## ## ## 0#####.)

### ##### ## ##### ## ##### ## #####
##### ## #####.#####. ##
##### ## ##### load, ##### ##
#####. ## ##### ## ## ## ##
DataResource ### ## ## ## #####.

##### ## ##### ## ##### 1.3, ##### [39]. ##
## ## ## ## ## ##. ## ##, ## ##
##_##### ## ## ##, (#####), ##
##### ## ## ## ##, ## ## #####.

```



```

## ##### ## ##### ## ##### ##, ## ##### ##### #####. ## ##
##### ##### ## ##### ##### ## ##### ##### #####
##### ## ##### ##### #####. (## #####, ## #####
##### ## ##### ## ## ##### ## ##### ## ## #####) ## ##### ##
##### ##### ##### ## ## #####, ##### ## ##### ##
##### ## #####.

```

#### 1.5.4.4. Sharing Resources among Annotators

```

##### ##### ## ## ##### ##### ## ##### ## ##### ## ##
##### #####. ## ##### ##### ## ##### ##### #####
#### ## ## ##### #####. ## ##### ##### #####
##### ##### ## ## ##### ##### ## ##### #####
## ##### ##### ## ##### ## ##### ## ##### ## ## ##
##### ## ##### ## ##### ##### ## ##### ##### ##
## ##### ##### ## ## ##### ## ##### #####, ## #####, ## ## ##
## ##### ## ##### ## ##### ##### ##### ##### ## #####

```

```

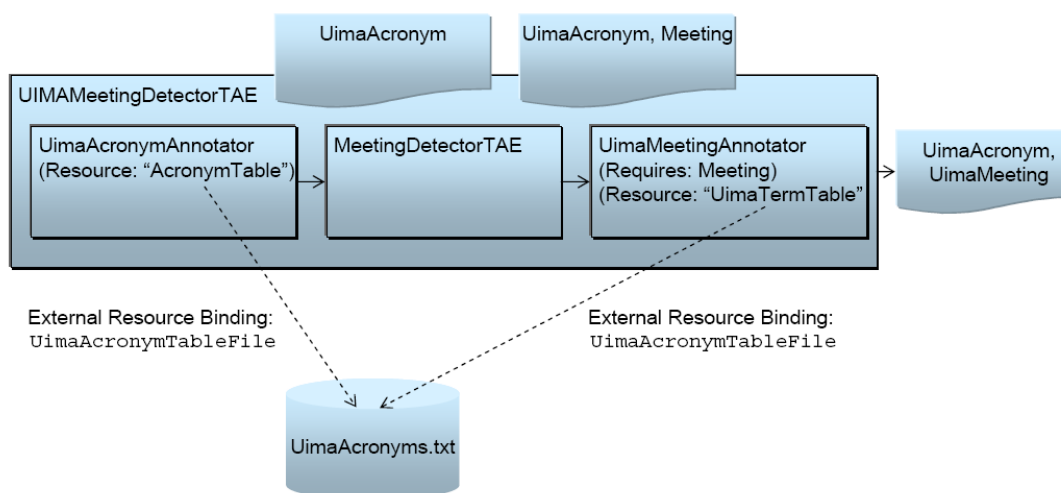
### ## ## ## ## ##### ## ##### 6 ##### # ## ##,
#####, ## ## ##### # ##### ##### 50 ##### ## ## ##

```

```

## ##### ##### #####, ## ##### ## #####
#####, ## ##### ##### ##### 4 ## ##### ##
##### ## ##### ## ##### #####. ##### ##
##### ## ## ## ##### #####, #####. ##
##### ## ## ##### # ## ##### ## ##### ## ##### 1.4,
##### ## ## ##### # ##### [40].

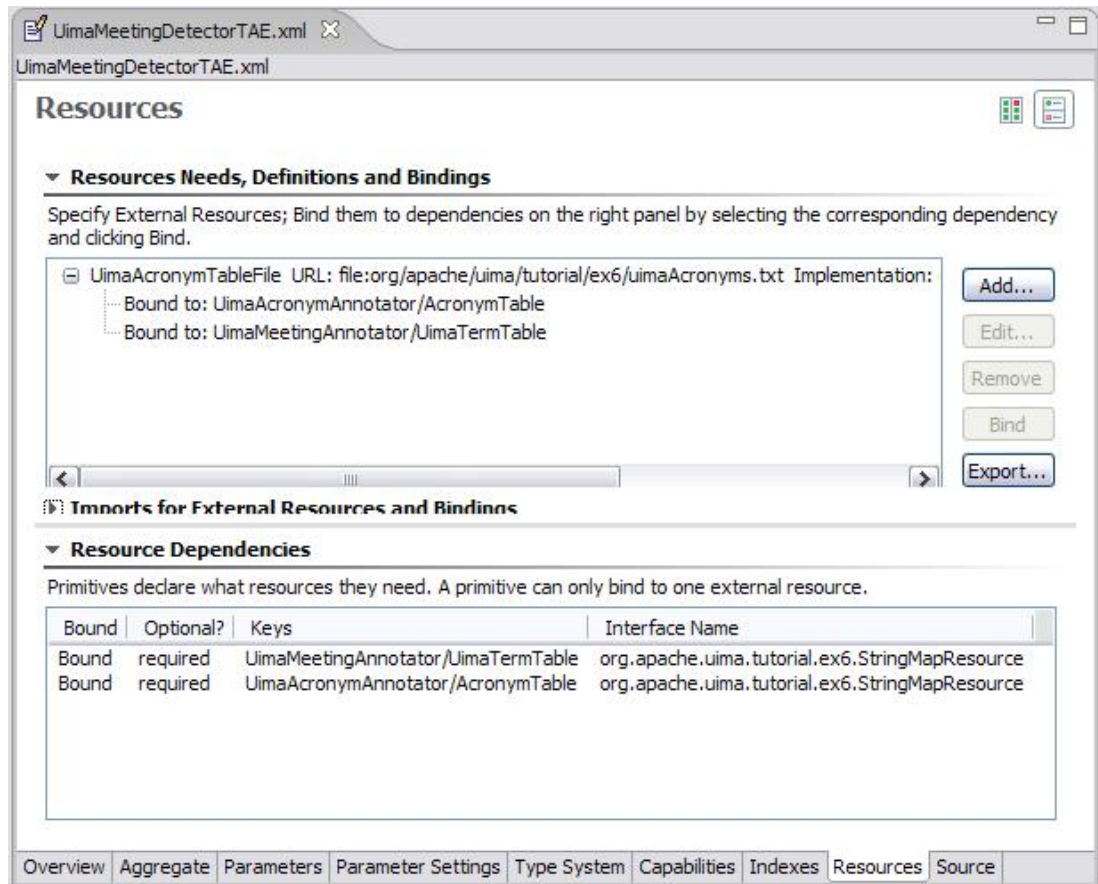
```



~~~~~

```
### ##### ##### ## ##### ## ## ## UimaMeetingDetectorAE.xml #####
#####. ## ##### ##### ## ##### ##### ##### ## ## #####
### ##### # ##### ##### ## ## ##### #####. (## #####
##### ## ## ##### ## ## ##### ## ##### ##### ## #####
##### ## ## ##### ## ## ##### ## ## #####).
```

#####:



```
<externalResourceBindings>
  <externalResourceBinding>
    <key>UimaAcronymAnnotator/AcronymTable</key>
    <resourceName>UimaAcronymTableFile</resourceName>
  </externalResourceBinding>

  <externalResourceBinding>
    <key>UimaMeetingAnnotator/UimaTermTable</key>
    <resourceName>UimaAcronymTableFile</resourceName>
  </externalResourceBinding>
</externalResourceBindings>
```

```
#### ##### ## ##### ##### ## ##### ##### (##### ##### ##
#####) ##### (#####) ## #####
#####. ##### ##### ## #####.
##### ## ##### override #####
#####.
```

```
## ## ##### ## ##### ##### #####, ## #####
## ####. ## ##### ## ##### ##### ## #####
## ##### ## #####. ##### ##### ##
##### ## ##### ##, #####.
```

1.5.4.5. Threading and Shared Resources

```
##### ## ##### ##### ##### ##### ##### ## ## ##### ## ##### ## ##
##### ## ##### ## ##### ##### ##### #####. ## ## #####
##### ## ##### ## ## ##### #####, ##### ## ##### ## #####
##### ## ##### ## # ##### #####, ## ## ##### ## #####. ##### ##
#####, ## ##### ## ##### ##### ##### ## ## ##### ## ## #####
# #### #, ## ##### ##### ##### ##### ## ##### ##### ## #####
##### ##### ##### ## ## ## ##### ## ##### ## #####, ##### ## #####
~~~~~ ~~~~~~ ~~~~~~ ~~~~~~ ~~~~~~ ~~~~~~ ~~~~~~ ~~~~~~ ~~~~~~ ~~~~~~.
```

```
## ## ##### ##### ## #####, ##### ## ##### ##### ##### #
##### #####, ##### ##### ##### ##### ##### ## ##### ##### ## (####
#####) ##### #####; ## ##### ## ## ## ##### ## ##### ##.
```

1.5.5. Result Specifications

```
## ##### ##### ## ##### ## ## ##### ##### ## ##### ##
##### #####. ##### ##, ##### ##### ##### ##
##### ##### ## ## ##### ##### ## ## ##### #####, ##### ## ##
##### ## ## ##### ##### ## ##### ## ##### ## #####.
```

```
# ##### ##### ## # ##### ## ##### ##### ## / ## #####:##### ##,
##### ## #####(##), ##### ## ##### ## ## ##### ## (##### ##) ##
#####. ##### ## ## ## ## ##### ##### #####, ##### ##, ##
##### ##### ## ## ##### ##### ## #####. ## ##### ## ## #####
##### ##### (## ## #####) ##### ##### ## ## ## #####
##### ## #####.
```

```
## ##### ##### ## ## ##### ##### ## ## ## ## ##
##### ## ##### #####; ##### ##### ## ## ##, ##
## ## ##### ##, ##. ##### ## ## # ##### ##, #####, ##### ##
##### ## ## ##### #####(##) ## #####.
```

```
##### ##### ## ## ##### ## ## ##### ##, ## ## ## ##
## #####. ## ## ##### ##### ## #####, ## ## ##### ## ## #####
##### ## #####. ##### ##### ## #####, ##
## #####: ## # ##### ## ## # ##### # ## ##### ##, ## ## ## ##
##### ##, ##, ## #####. ##### ## ##### ## ## ##: ## ##
##### ## ## ## # ## # ## ##### #####, ## ## ## #####
##### ## ## ##### (## #####, ##### ## #####).
```

```
## ##### ## ## ## ## ## ## ##### ##### ##### ## #####
##### ## #####, ## ## ##### ## ##### # ##### ## ##, ##
## ## ## ##, ## ##### ##, ##### ## ## (##### ## ##
#####) ##### #####. ##### ## ## ## ## ## ##### ## ##. #####, ## ##
##### ##### ##### ## ##### ##### ## ##, ## ## ## ## ##
```



```
##### ## ## ##### ## ##### #####; #####, ## ##### (##
#####, ##### # ##### ##### #####, ## #####). #### #####
##, ## ##### ## ## ##### ## (## #####, ##### ## ## ## #####(...)) #### ##
## ##### ##### ##### ##### ## ##### #####), # #####
##### ## #####.
```

1.5.5.1. Default ResultSpecification

#####.
/ ## #####, ### ##### ## ##### #####. ##
##, ### ##### ## ##### ## #####. #### ##
#####(##)#####; ### ##### #####
#####(##), ## ##### ##### #####
#####. ### ##### ## #####
(#####) # #####, ### #####, # #####. ## ##
#####, ## #####
#####.

1.5.5.2. Passing Result Specifications to Annotators

```
#####  
#####  
#####(#)  
AnalysisEngine.setResultSpecification(ResultSpecification) #####.  
  
#####  
AnalysisEngine.process(CAS, ResultSpecification).#####,###  
#####  
##### process.#####  
AnalysisEngine.setResultSpecification(ResultSpecification) #### ##  
#####.  
  
#####,  
##### setResultSpecification(ResultSpecification) #####.  
#####.
```

1.5.5.3. Aggregates

```
### #####, ### ##### ##### ## ##  
AnalysisEngine.setResultSpecification(ResultSpecification) ##### ##  
##### ## ##### ## ## ## #####/##### #### ## #####  
#####.#### ## ## ##### ##### ## ## ## ## #####/#####  
#### #### #####.### ####, ## ##### ## #### ## ##### ##  
##### ## ## ## ##### # #####, #### ##### ##  
##### ## ## ## ## #####.
```

```
## ##### #####, ###
AnalysisEngine.setResultSpecification(ResultSpecification) ## ##### ## ##
#####, ## ##### ##### ## ##### ## ##### ##### #####
## ## ## ##### ##### ## ##### ## ##### #####
#####. ##### ## ##### ## ## ##### ## ##### ##
##### ##### ## #####. ##### ##### ##### ## #####
## ## AnalysisEngine.setResultSpecification(ResultSpecification) ## ####
#####. ## ## ## ##### #####, ##### ## #####
#####.
```

1.5.5.4. Collection Processing Engines

```
### ##### ##### ## ##### ## ## ##### ## # #####
#####.
```

1.5.6. Class path setup when using JCas

```
#### ##### ##### ##### ## ##### ## ## #####. #####
##### ## ##### ## ## ##### ##### (##### ## ## #####)
##### #####).
```

```
### ##### ##### ##### ## ## ##### ##### ## #####
##### ## # ## ##. ##### ## ## ## ## ##### ## ## ##
#####.
```

```
### ##### ## ##### ##### ## ##### ##, #####
##### ##### ##### ## ##### ## ## ##### #####
# ##### ##### #####, ##### ## ##### 5.6.6, ##### ## #####
## #####.
```

1.5.7. Using the Shell Scripts

```
### ## ##### # /bin ##### ##### #####, ## ##### (.## #####)
### ## (.## #####). ##### ## ##### ##### ##### #####
# ##### ##; ##### ## ## ##### ##, setUimaClassPath ## ## ## ##
##### ##### ## ##### ## ## #####.
```

```
## ## ## ## ##### ## ## ## ##, ## ##### ## ## #####
##### ## ## ##### ##### ## ## _##### ## ## #####.
##, ## #####, ## ## ## ##### ## ##### #####, ## ##### ## ## ##
##### ## ## ## (## #####) #:#####.##, ## ##### #####
# set ##### ## ## ## _##### ## ## ##, ##### ## ## #####
#####:
```

```
set UIMA_CLASSPATH=c:\a\b\c\myProject\myJarFile.jar
documentAnalyzer
```

#####	#####
####_####	#### ##### ### ##### ### ### #####.
####_####	(#####) ##### # # ##### ##### #####. ## ### ###, ### ##### ### #### ## ## ##### ##### ## #####.
####_#####	(#####) ## #####, # ##### ##### ## ## ## ##### #####. ### ### ##### ### ##### #####. ## ### ### ####, ##### ##### ## #####.
####_#####	(#####) ## #####, # ##### ##### ## ## ## ##### ##### (### ##### 2.2, ##### ## ~~~~~)
####_#####_#####_####	(#####) ## #####, # ##### # # ##### ##### (### ##### 1.2, ##### [15])
####_###_####	(#####) ## #####, ### ##### ## ## ##### ## ## ## ##### #####. ##### ## ## ## ## ## ## ## ##### ##### ## ## #####.
###_####	(#####) ## #####, ### ##### ## ##### ## ## ##### (###) (### ##### 3.6.5, ##### ##### ##### (###)# [102])
#####_####	(#####) ##### # # # # # # # ##### ##### (##. #####_#####)

```
##### ##### ## #### ##### ##### ##### ## #####()
```

```

### ##### ## ##### ##### ## ##### #####() #####. ### ##
##### ##### ##### ## ## ##### ##### ## ##### ## ##### ##
##### ## # ## #####. #####, ## ##### ##### # ##### ## # ##
##### ##### ##### (#.#. ## ##### ## # ##### ##### #####) #####
## ##### ## # ## ##### ##### ## ## #####() #####. ## ## ##, ## ##### ##
## #####.

```

```

##### ## ## ##### ##

```

```

##### ## ## ## ## ## ##### ## ## ## ##### ## #####
##### ## ## ##### #####. # ##### ##### ##### ## ## ##
##### ## ## ##### ## ##### ## ##### ##### #####. #####
##### ##### ## ##### ## ##### ## ## ## ##### ##### ## ## ##
##### ##.

```

```

## #####, ## ##### ## ## ##### ##### ##### ##### ##### ##
##### ##### ##### (#####, ##, #####, ##). ##### ## ##### #####
##### ##### ## ## # ##### ##### ##### ##### #####, #####
##### ## ##### #####. #####, ##### ##### ## ##### #####
##### ## ##### ## ##### ##### ## ##### #####.

```

1.7. Viewing UIMA objects in the Eclipse debugger

```

##### (## ## ##### 3.1 ## #####) ## # ## ##### ## #####
#####. #####, ## ##### ## ## ## # ##### ## ##### (#### ##
##### #####, ## ## #####, ##.) ##### ##### ## #####
#####. ## #####, ## ## # ##### ## # ##### ##### ## ## #####
#####, ## ## ##### ##### 1:

```

Variables		Breakpoints	Expressions
Name	Value		
this	org.apache.uima.tutorial.ex1.RoomNumberAnnotator@1a0d253b		
aJCas	org.apache.uima.jcas.impl.JCasImpl@3a99653f		
docText	UIT Seminar: Challenges in Speech Recognition\n August 8, 2003 10:30 AM - 11:30 A...		
matcher	java.util.regex.Matcher@25fe53c		
annotation	RoomNumber\n sofa: _InitialView\n begin: 203\n end: 209\n building: "Yorktown"\n		
addr	21		
jcasType	org.apache.uima.tutorial.RoomNumber_Type@5a6ce538		

```

##### ##### ## ## ## ## # 2 #####, ## ##### ##
##### ## ##### ## ## ## ## ## ##### #####. ## ## ## ##
##### ##### ##### ## ## ##### ##### #####:

```

Variables	Breakpoints
-----------	-------------

```

##### ## ## ##### ##### ## ## ##:

```

Variables		Breakpoints	Expressions
Name	Value		
+	this		org.apache.uima.tutorial.ex1.RoomNumberAnnotator@1a0d253b
+	aJCas		org.apache.uima.jcas.impl.JCasImpl@3a99653f
+	docText		UIT Seminar: Challenges in Speech Recognition\n August 8, 2003 10:30 AM - 11:30 A...
+	matcher		java.util.regex.Matcher@25fe53c
-	annotation		RoomNumber\n sofa: _InitialView\n begin: 203\n end: 209\n building: Yorktown\n
	+	▲ [0]	Features: [Lorg.apache.uima.cas.impl.DebugNameValuePair;@1126538
	+	▲ [0]	sofa: Sofa\n sofaNum: 1\n sofaID: "_InitialView"\n mimeType: "text"\n sofaArray: ...
	+	▲ [1]	begin: 203
	+	▲ [2]	end: 209
	+	▲ [3]	building: Yorktown
	+	▲ [1]	Covered Text: 20-043
	+	▲ [2]	SubAnnotations: Expand to show

1.8. Introduction to Analysis Engine Descriptor XML Syntax

```
#####
#####
#####
#####
```

```
#####
##### 1.1, ##### [2]. #####
#####
```

```
### ##### 2,
#####
#####
```

1.8.1. Header and Annotator Class Identification

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Descriptor for the example RoomNumberAnnotator. -->
<analysisEngineDescription xmlns="http://uima.apache.org/resourceSpecifier">
  <frameworkImplementation>org.apache.uima.java</frameworkImplementation>
  <primitive>true</primitive>
  <annotatorImplementationName>
    org.apache.uima.tutorial.ex1.RoomNumberAnnotator
  </annotatorImplementationName>
```

```
#####
##### <analysisEngineDescription>, #####
http://uima.apache.org/resourceSpecifier.
```

```
#####,<frameworkImplementation>,#####
org.apache.uima.java.#####,<primitive>,#####
```

```
### #####, <annotatorImplementationName>, #####  
#####  
#####  
#####
```

```
#####  
#####
```

#####.#####
~~~~~ ##### ##### ##### ##### ##### #
##, ## #####. ##### ##### ##### ##### ## # ##### ##, ##
#####. ## ## #####
#####, ## #####
#####.

UIMA Version 2.3.0

```
### ##### ## ### ##### ##### ## ~~~~~ ## ###
##### # ### #####/##### ## ##### (#####) ### ### #####/##### ##### #
##### (#####). ##### ## ### ##### ## ##### ##### ##### ## ###
##### ##### ##### ##### ##### #####.
```

```
### ##### ##### ## ### ##, ##### ## ### ##, #####:#####.
### ##### (#####) ### #####.
```

```
### ##### ##### ## ##### ##### ##, ## ##### ##### #####
### # ##### ##### ## #####. #####, # ##### #####. ###
<#####> ##### ##### ## ##### #####:

```

```
<outputs>
  <type allAnnotatorFeatures = "true">
    org.apache.uima.tutorial.RoomNumber
  </type>
</outputs>
```

1.8.5. Configuration Parameters (Optional)

1.8.5.1. Configuration Parameter Declarations

```
<configurationParameters>
  <configurationParameter>
    <name>Patterns</name>
    <description>List of room number regular expression patterns.
    </description>
    <type>String</type>
    <multiValued>true</multiValued>
    <mandatory>true</mandatory>
  </configurationParameter>
  <configurationParameter>
    <name>Locations</name>
    <description>List of locations corresponding to the room number
      expressions specified by the Patterns parameter.
    </description>
    <type>String</type>
    <multiValued>true</multiValued>
    <mandatory>true</mandatory>
  </configurationParameter>
</configurationParameters>
```

```
### <configurationParameters> ##### ## ###
##### ##### ## ### #####. ## ### #####
### ##### ##, ## #####:

```

- ##### # ### ##### ## ##### ## ### #####
- ##### # # ##### ##### ## ### #####

- ##### # ### ##### ##### ## ### ##### ##### # ##### ## ### ## #####, #####, #####, ## #####.
- ##### # ##### ## ### ##### ##### ##### ##### (## #####), ##### ## ### ##### ##### ##### # ##### #####.
- ##### # ##### ## # ##### ##### ## ##### ##### ## #####

#####.

1.8.5.2. Configuration Parameter Settings

```
<configurationParameterSettings>
  <nameValuePair>
    <name>Patterns</name>
    <value>
      <array>
        <string>b[0-4]d-[0-2]ddb</string>
        <string>b[G1-4][NS]-[A-Z]ddb</string>
        <string>bJ[12]-[A-Z]ddb</string>
      </array>
    </value>
  </nameValuePair>
  <nameValuePair>
    <name>Locations</name>
    <value>
      <array>
        <string>Watson - Yorktown</string>
        <string>Watson - Hawthorne I</string>
        <string>Watson - Hawthorne II</string>
      </array>
    </value>
  </nameValuePair>
</configurationParameterSettings>
```

1.8.5.3. Aggregate Analysis Engine Descriptor

```
<?xml version="1.0" encoding="UTF-8" ?>
<analysisEngineDescription xmlns="http://uima.apache.org/resourceSpecifier">
  <frameworkImplementation>org.apache.uima.java</frameworkImplementation>
  <primitive>false</primitive>

  <delegateAnalysisEngineSpecifiers>
    <delegateAnalysisEngine key="RoomNumber">
      <import location=" ../ex2/RoomNumberAnnotator.xml" />
    </delegateAnalysisEngine>
    <delegateAnalysisEngine key="DateTime">
      <import location="TutorialDateTime.xml" />
    </delegateAnalysisEngine>
  </delegateAnalysisEngineSpecifiers>
```



```
### ##### ##### ##### ##### ##### ##### ##### ##### #####
## ##### <primitive> ##### ##### ##### false. ##### #####
##### ##### (##) ## ## ##### ## ##### ## # ##### ##.
```

```
####, ##### # # ##### ##### ####, ## ## # ## #
delegateAnalysisEngineSpecifiers. ##### ##### ## #
##### ## ##### . ## ##### ## ##### ## ##### #####
##### ## ## ##### ##### ## #####.
```

```
#### ## # ##### ## ##### ## ##### ## ##### ## #####
##### ## ## # ##### ##### ## ## #####:
```

```
<analysisEngineMetaData>
  <name>Aggregate AE - Room Number and DateTime Annotators</name>
  <description>Detects Room Numbers, Dates, and Times</description>
  <flowConstraints>
    <fixedFlow>
      <node>RoomNumber</node>
      <node>DateTime</node>
    </fixedFlow>
  </flowConstraints>
```

```
####, # ##### ## #####, ## ## ##### ## ##### ## #####
## #####. ## ##### ## ##### ## #####, ##### ## #####
##### ## ## ##### ## #####.
```

```
#####, ## ##### ## # #####, ##### ## ##### ## #####
## # ##### ## ## #####:
```

```
<capabilities>
  <capability>
    <inputs />
    <outputs>
      <type allAnnotatorFeatures="true">
        org.apache.uima.tutorial.RoomNumber
      </type>
      <type allAnnotatorFeatures="true">
        org.apache.uima.tutorial.DateAnnot
      </type>
      <type allAnnotatorFeatures="true">
        org.apache.uima.tutorial.TimeAnnot
      </type>
    </outputs>
    <languagesSupported>
      <language>en</language>
    </languagesSupported>
  </capability>
</capabilities>
```


Chapter 2. Collection Processing Engine Developer's Guide

[illegible]

(##). # ## #####
, #
 (##### ## ## ##### 2), ## . ##

, ## ##.

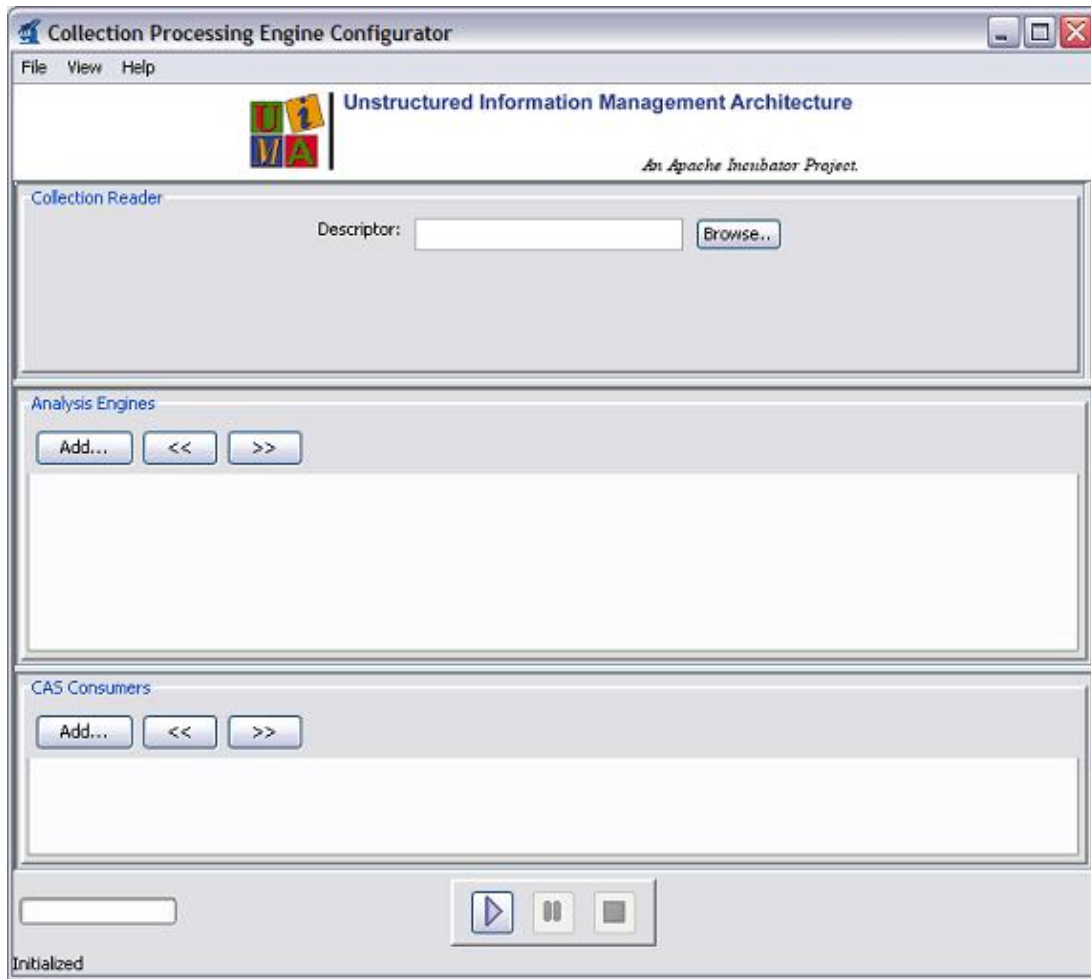
(#) #### . #
(#) #####
#####). #####. #####
, #
,
#####.
2.5. ##### [72].

[illegible]

¹ ### ##### ## ##### # ##### # # ##### #####, ##### ## #####.

- [illegible]

UIMA Version 2.3.0 CPE Developer's Guide 55



```
### ##### # ##### ##### #####, ### ##### #####, #####
#####, ### #####. 2 ## #####, ### ##### (#) ### ##
##### # ## # ##### # ##### # #####. ### #####
##### # ## # ##### # ## # ##### # ## #; ##### # #####
## ##### # ##### # ## # #####. ### #####, ### #####
##### # ## # ##### # ## # ##### # ## # #####:
```

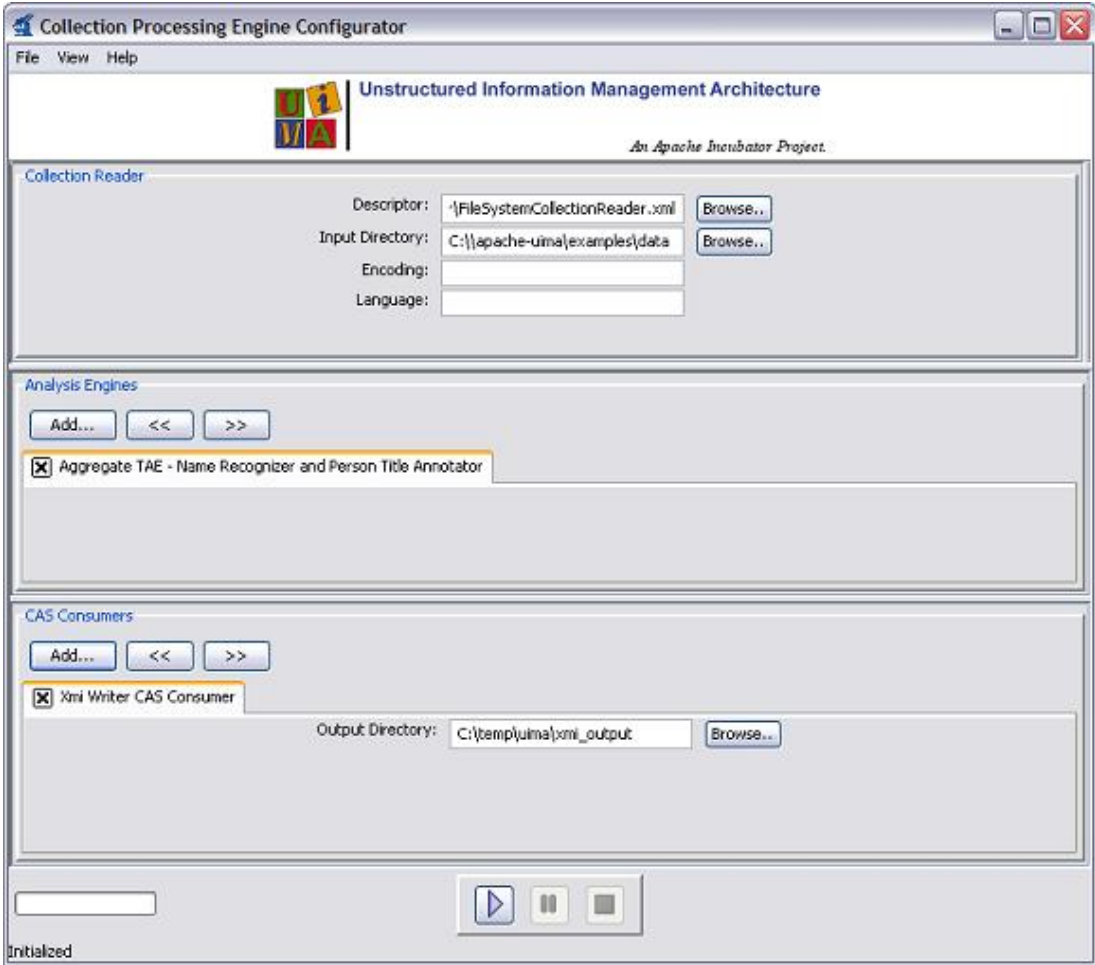
```
Collection Reader:
  %UIMA_HOME%/examples/descriptors/collection_reader/
  FileSystemCollectionReader.xml

Analysis Engine:
  %UIMA_HOME%/examples/descriptors/analysis_engine/
  NamesAndPersonTitles_TAE.xml

CAS Consumer:
  %UIMA_HOME%/examples/descriptors/cas_consumer/
```

²##### # ## # ##### # ## # ##### # ## # #####. ## ##### # ## # View → CAS Initializer Panel #####.

```
XmiWriterCasConsumer.xml
```



```
### ## ### ##### #####, ##### #### ## ##### ##### ## ## ##  
%UIMA_HOME%\examples\data3.### ##### ##### ## ## #####.### ## #####  
##### ## #####, ##### #### ## ##### ##### ## ## ## %UIMA_HOME%  
\examples\data\processed.
```

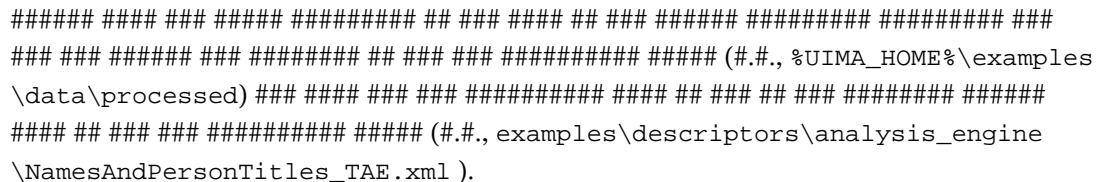

#####

#####

```
##### Save CPE Descriptor #####.###
##### Open
CPE Descriptor #####.###,###
```

³##### %UIMA HOME% ##### ## ##### ## ##### #####.

```
### ## ##### ## ## # ##### ## ## ##### ## ## #####
##### ## ## ## ## ## ##### ## ##### ##### ## ## #####
#####. ## ## ## #####, ##### ## ##### ## ##### ##
##### ## annotationViewer ##### ## (annotationViewer.bat ## #####,
annotationViewer.sh ## ##), ##### ## ##### ## ## bin ##### ## ## ## ##
#####. ##### ## ## ## ## ##### ## ##### ##### ##:
```



Analyzed Documents

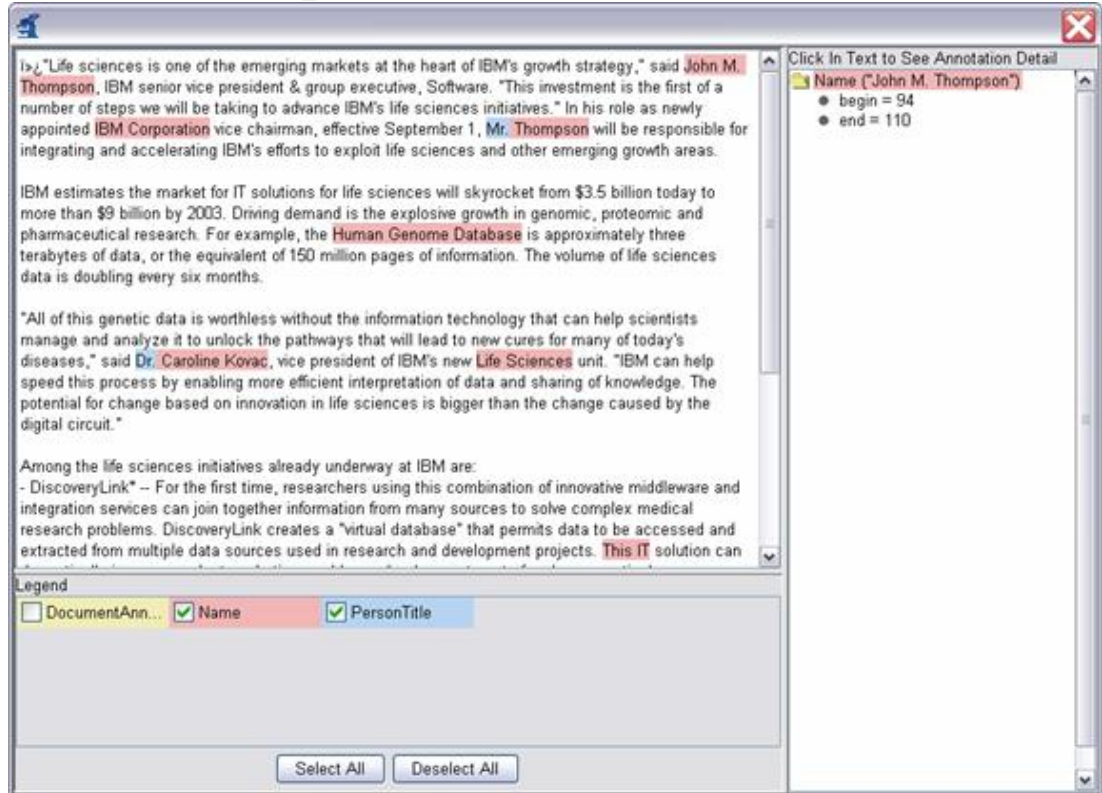
These are the Analyzed Documents.
 Select viewer type and double-click file to open.

- IBM_LifeSciences.txt
- New_IBM_Fellows.txt
- SeminarChallengesInSpeechRecognition.txt
- TrainableInformationExtractionSystems.txt
- UIMASummerSchool2003.txt
- UIMA_Seminars.txt
- WatsonConferenceRooms.txt

Results Display Format: ☒ Java Viewer ☐ HTML ☐ XML

Close


```
##### ## ## ##### ## ## ##### ## ##### #####. ##### #####
### #####, ### #####.###, ### ##### ## ## ##### #####:
```



```
#### ##### ## ## ##### ##### ## ## #####. ##### ## ## #####
##### ## ## ##### ## ## ##### ## ## ##### ## ## #####
####. ##### ## ## ##### #####. ##### ## ## #####.
```

```
#####! ### ##### ##### # ##, ##### ## #####, ## ##
##, ## ##### ## #####.
```

2.2.2. Running the CPE Configurator from Eclipse

```
## ## ##### ## ##### ## ##### 3, ~~~~~
~~~~~ ## ~~~~~
~ ~~~~~ ## ##### ##### #####, #### ## #####
##### ## ## ##### ## ## ##### ##### (##### UIMA CPE GUI)
##### ## ## ## #####. ##### ## ##### ## ##### ##
## #####.
```

```
## ## ##### ## ##### ## ##### ## ## ## ## ##
##### ## ## #####, ## ## ## ## ## ##. ## #####, ####
##### ##### ## ##### ## ## ##### #####. ##
##### ## ##### ## ##, ## ## ## ## ## ## ## ## ##
#####. ##### ## %####_####%/#####/#####. ##### ## ## ##
```

```
#####  
#####  
#####
```

```
####,###  
#####
```

```
####: #  
##  
#####  
#####  
#####  
#####  
#####  
#####  
#####
```

```
####,##  
#####
```

```
##  
##### org.apache.uima.tools.cpm.CpmFrame
```

```
##  
#####:
```

```
-Xms128M -Xmx256M  
-Duima.home="C:\Program Files\Apache\uima"
```

```
((  
#####)
```

```
#####  
#####
```

2.3. Running a CPE from Your Own Java Application

```
#####  
#####  
#####
```

```
//parse CPE descriptor in file specified on command line  
CpeDescription cpeDesc = UIMAFramework.getXMLParser().  
    parseCpeDescription(new XMLInputSource(args[0]));  
  
//instantiate CPE  
mCPE = UIMAFramework.produceCollectionProcessingEngine(cpeDesc);  
  
//Create and register a Status Callback Listener  
mCPE.addStatusCallbackListener(new StatusCallbackListenerImpl());  
  
//Start Processing  
mCPE.process();
```

```
#### ##### ##### ## ## ##### ## # ##### #####.
```

```
####: ## process() ##### ## # ## ## ## ## ##### ## ## ## ## ##
#### ## ## ##, ## ## ## ## ##### # ## ##, ## ## ## ## ## ## ## ##
#####.
```

2.3.1. Using Listeners

```
##### ## ## ##### #####, ##### ## ##### ## ## ##, ## ## ## ## #####
##### ## ## ##### ## ## ## ## addStatusCallbackListener, #####. ##
##### ##### ## # ##### ## ## ##### ## ## ## StatusCallbackListener
#####. ## ##### ## ##### ## ##### ##### ## ## #####. ## #####
#### ## ##### ##### ## ## ## ##### ## ## ##### # ## ##
org.apache.uima.examples.cpe.SimpleRunCPE.java ## ## %UIMA_HOME%\examples
\src ##### ## ## ##### ##.
```

```
## ## ## ## ##### ## ## ## ##### ## ## ## ## ##, ## ## ## ##
##### ## ## ## ##. ## ## ## ## ## ## ## ## ## org.apache.uima.collection
## ## ## ##.
```

2.4. Developing Collection Processing Components

```
#### ##### ## ## ##### ## ## ## ## ## ##### ## ## ##, ##
##### ## ## ## ##. ## ## ## ## ## ## ## ## ## ## ## ##
##### ## %UIMA_HOME%\examples\src ##### ##.
```

```
## ## ##### ## ## ##, ##### ## ## ## ## ##### ## ## ## ##
## ## ## ##, 0##### ## ## ## ## ## ## ## ## ## ## ##
#####. (##### ## ## ## ## ## ## ## ## ## ## ## ## ## ##, ##
#####, ## ## # 0##### ## ## ## ## ## ## ## ## ##, # ## ## ## ##
#### ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## 0##### ## ## ## ##.)
```

2.4.1. Developing Collection Readers

```
# ##### ## ## ##### ## ## ##### ## ## ## ## ## ## ## ## ##
## ## ##### ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##
##### ## ## ## ## # ## ## ## ## ## ## ## ## ## ##.
```

```
# ##### ## # ##### ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##
##### ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##
##### org.apache.uima.examples.cpe.FileSystemCollectionReader ## ## ##
##### ## %UIMA_HOME%\examples\src\main\descriptors\collection_reader\
FileSystemCollectionReader.xml.
```

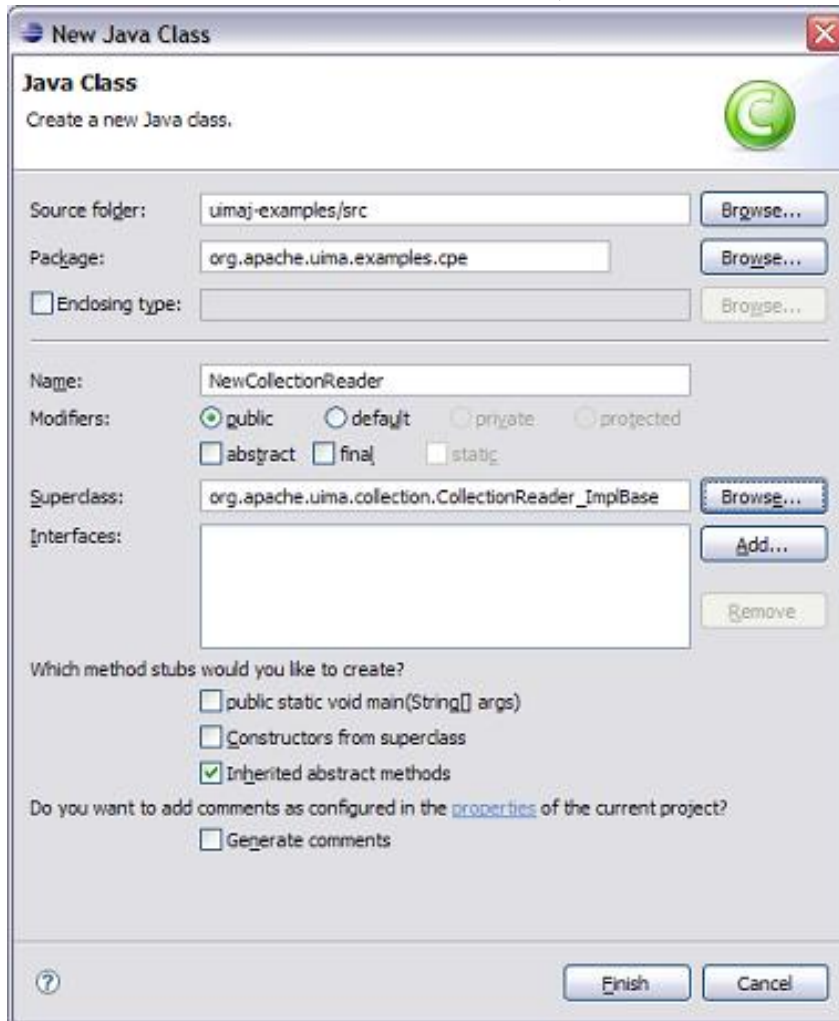
2.4.1.1. Java Class for the Collection Reader

```
## ## ## ## ## # ##### ## ## ## ## ## ## ## ## ## ##
org.apache.uima.collection.CollectionReader #####. ## ## ## ## ##
```

```
##### ##### ##### ##### ##### #####, ## ## ## ##### ##
##### ##### ##### org.apache.uima.collection.CollectionReader_ImplBase .

##### ##### ##### ##### ##### ##### ##### ##### ##### #####
##### ## ## CollectionReader #####, ## ##### ##### ##### ##
##### ##### ##### ## ##### ## ##### ## ##### ##### #####. #####
##### ## ## ##### ##### #####, ## ## ## ##### ## ##### ##### #####
##### ##### ##### CollectionReader #####.

###: ##### ## # ## ## ## ##### #####, ## ## ##### ##### ##
##### ##### ##### ## ## ## ## ## ## ## ##### ## #####
File → New → Class ## ##### ## ## ## ## ##### #####, #####
org.apache.uima.collection.CollectionReader_ImplBase ## ##
#####, ## ##### ##### ##### ##### ##### ## ## #####
##### ##### ##### ## ## ## #####?#, ## ## ## ##### #####:
```



```
##### ## ## ## ##### ## ##### ##### ##### #####
##### ## ## CollectionReader_ImplBase #####, ## ## ## ##### ##
org.apache.uima.examples.cpe.FileSystemCollectionReader . ## ## ## #####
```

```
#### # #####,### ##### ##### ##### #####
### CollectionReader ##### # ## ##### CollectionReader ## ##
#####.
```

2.4.1.2. Required Methods in the Collection Reader class

```
### #####:

```

initialize()

```
### initialize() ##### # ##### # #####
## #####.CollectionReader_ImplBase ##### # #####
##### (#.,## # #####),## # ##
##### # #####. #####, # #####
##### # #####
##### # #####

## # ##,### ##### # ##### # ##### # #####
##### # #####.### # #####
##### # ##### # ##### # ##### # #####
##### # #####:
```

```
public void initialize() throws ResourceInitializationException {
    File directory = new File(
        (String)getConfigParameterValue(PARAM_INPUTDIR));
    mEncoding = (String)getConfigParameterValue(PARAM_ENCODING);
    mDocumentTextXmlTagName = (String)getConfigParameterValue(PARAM_XMLTAG);
    mLanguage = (String)getConfigParameterValue(PARAM_LANGUAGE);
    mCurrentIndex = 0;

    //get list of files (not subdirectories) in the specified directory
    mFiles = new ArrayList();
    File[] files = directory.listFiles();
    for (int i = 0; i < files.length; i++) {
        if (!files[i].isDirectory()) {
            mFiles.add(files[i]);
        }
    }
}
```

```
####:### # ##### # #####
##### # ##### # #####
initialize(ResourceSpecifier, Map) ## # ##
##### # #####.#####
##### # ##### initialize().
```

hasNext()

```
### hasNext() ##### # ##### # #####
##### # #####.### # ##### hasNext() #####
##### # ##### # ##### # ##### # #####:
```

```
public boolean hasNext() {
    return mCurrentIndex < mFiles.size();
}
```

getNext(CAS)

```
### getNext() #####
# ###. ## ### #####, ##### ## ##### ## ##### ## #####
setDocumentText #####. ### ##### ##### ##### ##### ## #####
#####. ## ##### ## # ## #####. ## ## ## ##### # ## #####,
### ## ##### ## ## ## ## ## ## ## initialize() ## ##. ## ##
### ## ## ##### # ## #####, ## ## ## ##### ##### ## ##
##### ## ## ## ##### ## ## ## ##.

### ## ## ##### ##### ## ## ##### ##### ## ## #####
## ## ##. ## #####, ## ## ## ##### ## ## ##### #####
##### ##### uima.tcas.DocumentAnnotation (### ##### 4.3, ##### ##
##### ## ~~~~~ ## ## ##### #####) ##
##### ## ##### ## org.apache.uima.examples.SourceDocumentInformation
, ##### ##### ##### ## ##### #####. #####
##### ## ## ## ## ##### ## ## ## ##.
#### ## ## ## ## ##### ## ## ## ## ## ## ##
org.apache.uima.examples.SourceDocumentInformation.xml, ##### ## ##### ##
### examples/src #####.

### #####() ##### ## ## ## ## ##### ##### ##### #####:
```

```
public void getNext(CAS aCAS) throws IOException, CollectionException {
    JCas jcas;
    try {
        jcas = aCAS.getJCas();
    } catch (CASEException e) {
        throw new CollectionException(e);
    }

    // open input stream to file
    File file = (File) mFiles.get(mCurrentIndex++);
    BufferedInputStream fis =
        new BufferedInputStream(new FileInputStream(file));
    try {
        byte[] contents = new byte[(int) file.length()];
        fis.read(contents);
        String text;
        if (mEncoding != null) {
            text = new String(contents, mEncoding);
        } else {
            text = new String(contents);
        }
        // put document in CAS
    }
```

```

        jcas.setDocumentText(text);
    } finally {
        if (fis != null)
            fis.close();
    }

    // set language if it was explicitly specified
    //as a configuration parameter
    if (mLanguage != null) {
        ((DocumentAnnotation) jcas.getDocumentAnnotationFs()).
            setLanguage(mLanguage);
    }

    // Also store location of source document in CAS.
    // This information is critical if CAS Consumers will
    // need to know where the original document contents
    // are located.
    // For example, the Semantic Search CAS Indexer
    // writes this information into the search index that
    // it creates, which allows applications that use the
    // search index to locate the documents that satisfy
    //their semantic queries.
    SourceDocumentInformation srcDocInfo =
        new SourceDocumentInformation(jcas);
    srcDocInfo.setUri(
        file.getAbsolutePath().toURL().toString());
    srcDocInfo.setOffsetInSource(0);
    srcDocInfo.setDocumentSize((int) file.length());
    srcDocInfo.setLastSegment(
        mCurrentIndex == mFiles.size());
    srcDocInfo.addToIndexes();
}

```

```
#####  
#####
```

getProgress()

```
#####  
#####;###  
#####  
## Progress ##  
##### (#), ## (#.#) (##),  
#####, ##). #####  
#####  
##### getProgress() ####:
```

```
public Progress[] getProgress() {
    return new Progress[]{
        new ProgressImpl(mCurrentIndex,mFiles.size(),Progress.ENTITIES)};
}
```

```
## ##### #####, ## ##### ##### ## ##### ## ##### ##### ## #####, ##
## ##### ##### ## ##### ##### ## #####. ## #####, # ProgressImpl ##### ##
Progress.ENTITIES ## #####, ## # ProgressImpl ##### ## Progress.BYTES ##
##.
```

close()

```
## ##### ##### ## ##### ##### ## ##### ##### ## ## ##### #####. ##
##### ##### ##### ##### ##### ## ##### ## ## ## #####. ##
##### ##### ##### ##### ##### ## ## ## ## ##### #####
## #####:
```

```
public void close() throws IOException { }
```

Optional Methods

```
## ##### ##### ## ## #####:
```

reconfigure()

```
#### ##### ## ##### ## ## ##### ##### ##### ##### #####.
```

typeSystemInit()

```
## ## ## ##### ##### ##### ## ## ##, ## ## ## ##### ## ##
(#####, ## ## ## ##### #####, ## ## ## ##### ## ##### #####. ## ##
## ##### ##### ## ## ##, ##### ##### ## ## ## ## ## ## ## ## ##
##### # ## ##### 1.5.1, ##### ##### \[29\] ## #####.
```

Threading considerations

```
##### ##### ## ## ## ## ## ## ##; ##### ## ## ## ## ## ##### ##
#####, ## ## ## ##### ## ##### ## ## ##### ##### ##### (##)
## ##.
```

XML Descriptor for a Collection Reader

```
## ## ## ## ##### ##### ##### ## ##### ## / ## ##### ## #####
##### ##### #####. ##### ## ## ##### (#####), #####
## ##### ## ## ##### #####:
```

```
<collectionReaderDescription
  xmlns="http://uima.apache.org/resourceSpecifier">
  <frameworkImplementation>org.apache.uima.java</frameworkImplementation>
  <implementationName>
    org.apache.uima.examples.cpe.FileSystemCollectionReader
  </implementationName>
  <processingResourceMetaData>
    <name>File System Collection Reader</name>
    <description>Reads files from the filesystem.</description>
    <version>1.0</version>
```



```

<vendor>The Apache Software Foundation</vendor>
<configurationParameters>
  <configurationParameter>
    <name>InputDirectory</name>
    <description>Directory containing input files</description>
    <type>String</type>
    <multiValued>>false</multiValued>
    <mandatory>>true</mandatory>
  </configurationParameter>
  <configurationParameter>
    <name>Encoding</name>
    <description>Character encoding for the documents.</description>
    <type>String</type>
    <multiValued>>false</multiValued>
    <mandatory>>false</mandatory>
  </configurationParameter>
  <configurationParameter>
    <name>Language</name>
    <description>ISO language code for the documents</description>
    <type>String</type>
    <multiValued>>false</multiValued>
    <mandatory>>false</mandatory>
  </configurationParameter>
</configurationParameters>
<configurationParameterSettings>
  <nameValuePair>
    <name>InputDirectory</name>
    <value>
      <string>C:/Program Files/apache/uima/examples/data</string>
    </value>
  </nameValuePair>
</configurationParameterSettings>

<!-- Type System of CASes returned by this Collection Reader -->

<typeSystemDescription>
  <imports>
    <import name="org.apache.uima.examples.SourceDocumentInformation" />
  </imports>
</typeSystemDescription>

<capabilities>
  <capability>
    <inputs/>
    <outputs>
      <type allAnnotatorFeatures="true">
        org.apache.uima.examples.SourceDocumentInformation
      </type>
    </outputs>
  </capability>
</capabilities>
<operationalProperties>
  <modifiesCas>true</modifiesCas>
  <multipleDeploymentAllowed>>false</multipleDeploymentAllowed>
  <outputsNewCASes>true</outputsNewCASes>
</operationalProperties>
</processingResourceMetaData>

```


XmiCasSerializer #####
 ##### (#####).

```
public void processCas(CAS aCAS) throws ResourceProcessException {
    String modelFileName = null;

    JCas jcas;
    try {
        jcas = aCAS.getJCas();
    } catch (CASEException e) {
        throw new ResourceProcessException(e);
    }

    // retrieve the filename of the input file from the CAS
    FSIterator it = jcas
        .getAnnotationIndex(SourceDocumentInformation.type)
        .iterator();
    File outFile = null;
    if (it.hasNext()) {
        SourceDocumentInformation fileLoc =
            (SourceDocumentInformation) it.next();
        File inFile;
        try {
            inFile = new File(new URL(fileLoc.getUri()).getPath());
            String outFileName = inFile.getName();
            if (fileLoc.getOffsetInSource() > 0) {
                outFileName += ("_" + fileLoc.getOffsetInSource());
            }
            outFileName += ".xmi";
            outFile = new File(mOutputDir, outFileName);
            modelFileName = mOutputDir.getAbsolutePath() +
                "/" + inFile.getName() + ".ecore";
        } catch (MalformedURLException e1) {
            // invalid URL, use default processing below
        }
    }
    if (outFile == null) {
        outFile = new File(mOutputDir, "doc" + mDocNum++);
    }
    // serialize XCAS and write to output file
    try {
        writeXmi(jcas.getCas(), outFile, modelFileName);
    } catch (IOException e) {
        throw new ResourceProcessException(e);
    } catch (SAXException e) {
        throw new ResourceProcessException(e);
    }
}
```

Optional Methods

#####

batchProcessComplete()

```
### ##### ##### ## #####() ##### ## ## ## ## ## ## ## ##
#####. ##### ## ## ##### ## ##### ##### ## ##### ## #####
### ##### #####. ### ##### ## ##### ## ##### ## ## ##### ##
##### ## #####, ## ## ## ## ## ##. ##### ## ## ## ## ## #####
##### ##### #####.
```

collectionProcessComplete()

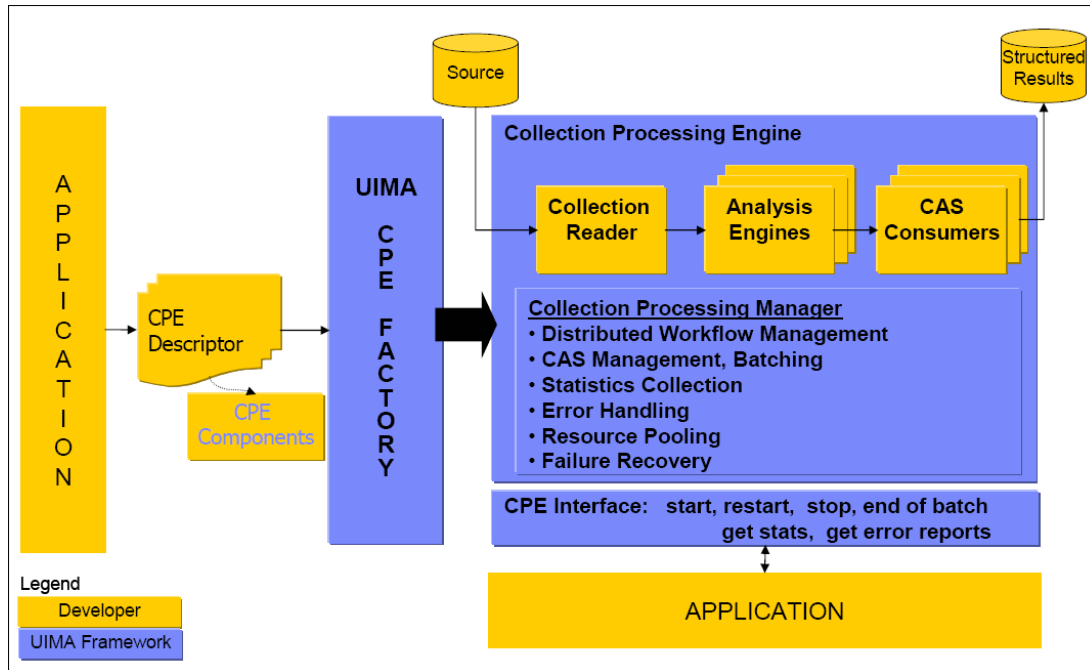
```
### ##### ##### ## #####() ##### ## ## ## ## ##
##### (#.#., ##### ## ## ##### ## ## #####). ## ## ## ##
## ## ##, ## ## ## ## ## ## ##. ##### ## ## ## ## ## ##
##### ## ##### ## ##### ##### ##### ## ## ## ## ## ##
## ## #####. ### ##### ## ##### ## ##### ## ## ## ##
##### ## ## ## ## ## ##.
```

2.5. Deploying a CPE

```
### ## ##### # ##### ## ##### ## ##### ##### #####
### ##### ## ## ##, ##### #####, ## ##### ## ##### ##### ## ## ##
#####. ### ##### ## ## ## (### #####, ## ##) ## ##### ##
##### ##### ## ##### ## ## ## ## #####. ### ##### ##### ## ##
### ##### ## ##, #####, ##### ## ## ##### ##### #####
#####. ## ##### ## ## ##, ## ## ##### ## ## ## ##.
```

```
##### ## ## ##### ## ## ## ##### ##### ##### ## #
##### ##### ## ## ## ##### ## ## #####. ## ## #####, ## #####
#### # #####, ##### ##### ## ##### ##### ## ## ##
## #####, ## ##### ## ##### ## ## ##### 3, ~~~~~
~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ##
~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ~~~~~ ##
#####.
```

```
##### 2.2, ##### ##### [72] ##### # ##### ## ## ## ## #####
### ##### ## ##### ## ## ## ## ## #####. ### ## #####
##### ## ## ##### (##### ##### #####)
### ##### ## ##### ## ##### ## ## ## ## ##### ## ##
#####.
```



#####) ## # ###:

1. ##### (#### ## ### ##### ##### ## ### ##)
2. ##### (#### ## # ##### ##### ## ### ##### #####), ###
3. ##### (#### ## # ##### #####, ##### ## # ##### #####).

service ##

#####

####: ### #####

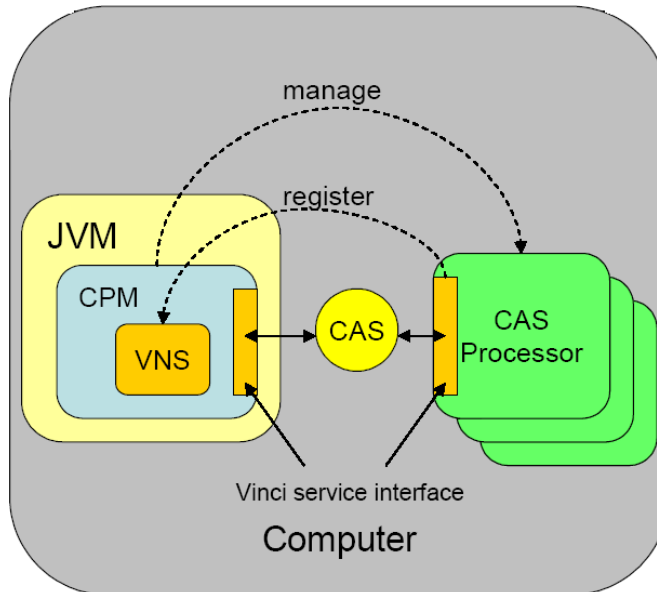
[96]. #####
#####

```
### ##, ## ## ##### ##, #### ## ##### ## ## ##### ##, ##
####, ##### ## # ##### #####.
```

```
### ## ##### ## ##### ## ##### ## ##### ## ##### ##
##### ## ##### ##. ## ##### ## ##### ## ## ##### ##, ## ##
##### ## ## ##### ## ## (##### ## ## ##### ##). ##### ## ##
### ##### ## ## ##### ## ##### ## ##### ## ##### ##
### ## ##### ## ##### 3, ~~~~~
~~~~~ ## ~~~~~ .## ##
##### ##### ## ##### ## ##### ## ##### ## ##### ##.
```

2.5.1. Deploying Managed CAS Processors

```
##### ## ##### ## ##### ## ##### ## ##### 2.3, ##### ##
##### [73]. # ##### ## ##### ## ##### ## ## ## ## # #####
### ## ##### ## ##### ## ## ## ##### ##### #####, #####
## #####, ## ##### #####. # ##### ## ##### ## ## ## ##
## ## ##, ## ## # ##### #####. ##### ## ##### ##### #####
### ## ## ##### ## ##### ##### ## #####. # ##### ## # #####
##### ##### ## ##### ## ##### ## ## ##.
```



```
~~~~~
### ## ##### ## ##### ## ##### ## #####
#####. # ## ##### ## ##### ## # ##### ## ## process() #####
## ##### ## ## #####. ## ## ## ## ##### ## ##
##### ## #####. ## ##, ## #####, ##### ## ## 9005. ## ## ##
#####, ## ## ## ##### ## ##### ## ##### ## ##### ##
### ##### ## ##### ## ##### ## ##### ## ## ##
#### ##### ## ## ##### ## ## ## ## ##### ## ##.
```

```
## ##### # ##### ### #####, ## ## ##### #### ##### ## ## #####. ##
##### ## # ##### #### ## ## ##### #### ##### ## ##### #####
##### # ##### ## #####.
```

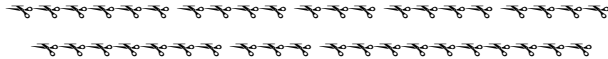
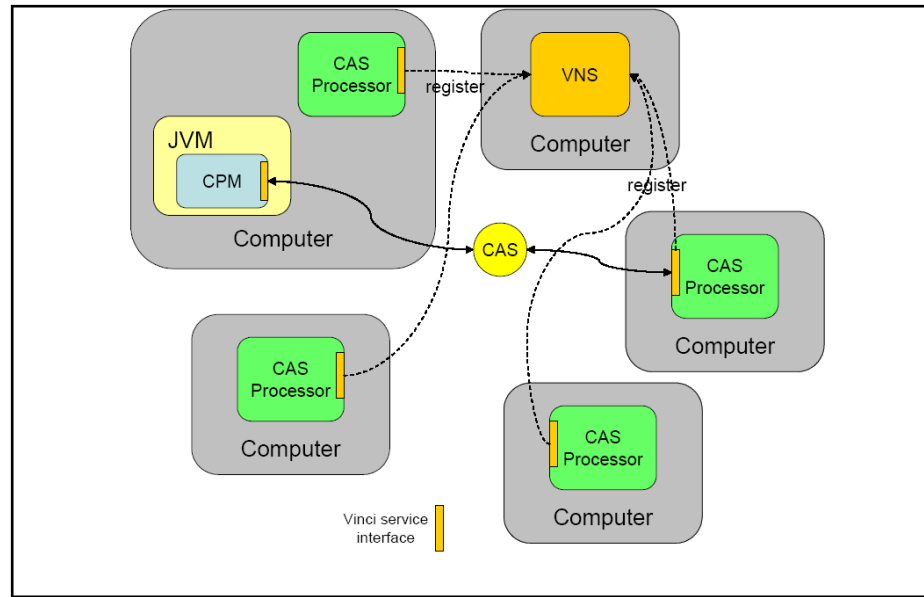
```
<casProcessor deployment="local" name="Meeting Detector TAE">
  <descriptor>
    <include href="deploy/vinci/Deploy_MeetingDetectorTAE.xml"/>
  </descriptor>
  <runInSeparateProcess>
    <exec dir="." executable="java">
      <env key="CLASSPATH"
        value="src;
          C:/Program Files/apache/uima/lib/uima-core.jar;
          C:/Program Files/apache/uima/lib/uima-cpe.jar;
          C:/Program Files/apache/uima/lib/uima-examples.jar;
          C:/Program Files/apache/uima/lib/uima-adapter-vinci.jar;
          C:/Program Files/apache/uima/lib/jVinci.jar"/>
      <arg>-DLOG=C:/Temp/service.log</arg>
      <arg>org.apache.uima.reference_impl.collection.
        service.vinci.VinciAnalysisEngineService_impl</arg>
      <arg>${descriptor}</arg>
    </exec>
  </runInSeparateProcess>
  <deploymentParameters/>
  <filter/>
  <errorHandling>
    <errorRateThreshold action="terminate" value="1/100"/>
    <maxConsecutiveRestarts action="terminate" value="3"/>
    <timeout max="100000"/>
  </errorHandling>
  <checkpoint batch="10000"/>
</casProcessor>
```

```
## ##### 3, ~~~~~
~~~~~ ## ~~~~~
## #####
#####.
```

2.5.2. Deploying Non-managed CAS Processors

```
##### ## ##### ##### ## ##### ## ##### 2.4, ##### ##### ##
##### [75]. ## ##### ##, ## ## ##### ##### ## ## #####
##### ## ##### ## ##### #####. ##### ##### ## #####
#### ##### ## ## #####.
```

1. ##### ## ## ##.
2. ##### ## ##, ##### ## ## ## ## ##.



#####

3.6.5.1, ##### [102].

#####

```
<casProcessor deployment="remote" name="Meeting Detector TAE">
  <descriptor>
    <include href=
      "descriptors/vinciService/MeetingDetectorVinciService.xml" />
    </descriptor>
    <deploymentParameters/>
    <filter/>
    <errorHandling>
      <errorRateThreshold action="terminate" value="1/100" />
      <maxConsecutiveRestarts action="terminate" value="3" />
      <timeout max="100000" />
    </errorHandling>
    <checkpoint batch="10000" />
  </casProcessor>
```

```

### ##### 3, #####
##### ##
##### #####.

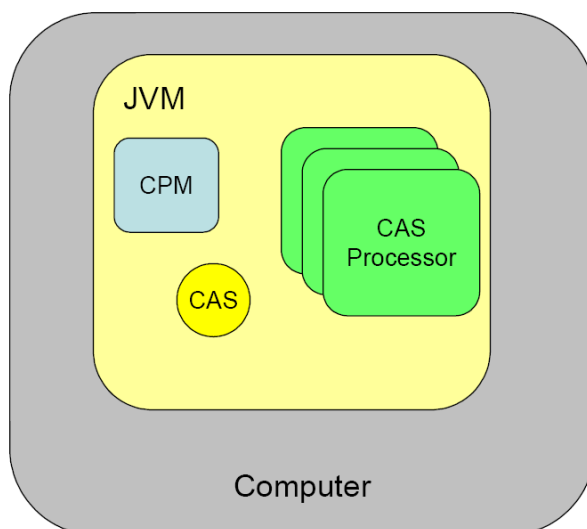
```

2.5.3. Deploying Integrated CAS Processors

```

##### ## ##### ## ##### # ##### 2.5, ##### ##### ##
##### [76]. ##### ## ##### ## ## ## ##, ##### ##
##### ##### ## ## #####. ##### ##### ## #####
##### ##### ## ##### ##### ## ## ## ## ##### ## #####
##### ## ## ##. #####, # ## ##### ##### ## ##### ## #####
## ##### ## ## ##### ## ## ##### ##### ## ##### ## #####.
##### ## ##### # ##### ##### ##### ##### ##### ##### # #####
##### ## ##### ## ##### ## ## ##, ##### ## ##### ##, ## #####.

```



```

##### ## # ##### ##### # ## ##### ##### ##### ## #####
##### ##### ## #####.

```

```

<casProcessor deployment="integrated" name="Meeting Detector TAE">
  <descriptor>
    <include href="descriptors/tutorial/ex4/MeetingDetectorTAE.xml"/>
  </descriptor>
  <deploymentParameters/>
  <filter/>
  <errorHandling>
    <errorRateThreshold action="terminate" value="100/1000"/>
    <maxConsecutiveRestarts action="terminate" value="30"/>
    <timeout max="100000"/>
  </errorHandling>
  <checkpoint batch="10000"/>
</casProcessor>

```

```
### ##### 3, ~~~~~
~~~~~ ## ~~~~~
### #####.

```

2.6. Collection Processing Examples

```
### ##### # ## # ##### ##### ## #####,
#####,,### #####. ##### ## ## /examples/descriptors/
collection_processing_engine #####. ##### ## ##### ##### ##
## ##### ##### (## #####) ## #####.

```

```
## ## ##### ## ##### ## #####, ## ## runCPE ##### ## ## /##
##### ## ## #####, ##### ## ##### ## ##
#####,,## ## ##### ##### ## ## ## uimaj-examples ##### ## ##
#####,,## ## ## ## ## → ## → ##... → ## ## ## ## ##
##### ## ## ##.

```

```
####: ## runCPE ##### must ## ## ## ## %UIMA_HOME%\examples #####,
##### ## ##### ## ##### ## ##### ## ## ##
##### ## ## #####. ## ##,

```

```
#####
#####_#####_#####.##

```

```
## ## ## #####, ##### ## ## #####.

```

1. ##### # ## ##### ## ##### ## startVNS ##### ## ## /bin #####, ##
#####.
2. ##### ## ##### ##### ##### ## # ##### ##, ## ##### ##
startVinciService ##### ## ## /bin ##### ## ##### ##
##, ## ##### ## ## ##### ## ## ##### ## ##, ## ## ##
%UIMA_HOME%/examples/deploy/vinci/Deploy_MeetingDetectorTAE.xml,
uimaj-examples ##### ## ##
#####,,## ## ## ## ## → ## → ##... → ## ## ## ## ##
##.
3. ##, ## ## ##### ##### (## ## ## #####, ## ## #####
##), ##### ## ## ## ## ##
(%UIMA_HOME%/examples/descriptors/collection_processing_engine/
MeetingFinderCPE_NonManaged.xml).

```
#### ##### ## ## ##### #####, ## ##### ## ##  
MeetingDetectorTAE ##### ## ## ##### ## ## #####. ##### ## ##  
##### ## ## ##### ## ## ## ## ## ## ##  
##_#### ## ## _####; ##### ## ##### ## #9000#. ## ## ## ##  
##### ## ## ## ## ## ## ## ##, ## ## ## ## ## ## ##  
## ## ## ## ## ## ## ## ## ## ## ##.

```

```
#####,### ## #### ## ##### ##/## ## ## ##### ## ##### #####
### ## ##_#### ## ##_####.### #####,## ## runCPE ##### ##
##### ## # ##### ##### ## ## ##### #####,### ## #### ##
MeetingFinderCPE_NonManaged.xml ### ##### ## ##### #####: <parameter
name="vnsHost" value="localhost" type="string"/> ## ##### ## ## ##
##### ## #####.
```

Chapter 3. Application Developer's Guide

```
#####  
##### (###). ###  
# #####  
## #####, #####,  
#####,  
#####.
```

3.1. The UIMAFramework Class

```
#####  
## org.apache.uima.UIMAFramework #####.#####  
#####  
#####  
##### (#####  
#####).
```

- #####(): #####,
#####.
#####.
- #####(#####): #####

#####. #####,

#####.
- #####
- #####
- #####
- #####
- #####
- #####,
#####.
- #####(<#####>): #####
#####,
#####.
- #####(): #####
- #####(): #####
#####.

```

### ## ##### ## ##### ##### ##### ##### ## ## ## ## ##
##### #####. ##### ##### ##### ##### # #####,
### ## ## ## ## ## ## #####.

```

3.2. Using Analysis Engines

```

#### ##### ## ## ## ##### ##### ## ## ##### ##
##### ##### ##### ##### ## ## ##. ## ~~~~~
~~~~~ ## # ##### ##### ##### ##### (#.#. #####) ## #####
#####.

```

```

## ##### ##### ## ## ## # ##### (##### ## ## ##
##### ## #####) ## ~~~~~ (## ## ## ##). ## ## ##
## ## ## ## ## ## ##### ## ## ##, ## ##### ## ## ##
##### ##### ## ##. ## ## ##### ## ## ## ## ##
##### #####.

```

```

### ## ~~~~~ ~~~~~ ~~~~~ ~~~~~ ## ##### ##
##### ## ##### ##### ##### # #####. ## ## ## ## #1.#, #####
## # ##### ##### ## ## #####. #####, ## ## ## ## ##
#2.0, ## ## ##### ## ## ##### ## ## ##### ## ##
## #####.

```

```

### ## ##### ## ##### ##### ##### ##### ## ## #####
## ## ## # ##### ## ## ## ## ## #####. ## ## ## ##
##### ##### ## ## ## ##### ## ## ##### ## ##
##### ## ## ## ## ##### ##### (## ##### ## ## ##
#####) ##### ## ## ##.

```

3.2.1. Instantiating an Analysis Engine

```

### ##### ##### ## ## ##### ## ## ## ## ##:

```

```

//get Resource Specifier from XML file
XMLInputSource in = new XMLInputSource("MyDescriptor.xml");
ResourceSpecifier specifier =
    UIMAFramework.getXMLParser().parseResourceSpecifier(in);

//create AE here
AnalysisEngine ae =
    UIMAFramework.produceAnalysisEngine(specifier);

```

```

### ##### ## ##### ## ## ##### (## ## ## #####,
## ## ## ## ## ##### # ## ## ##### ##### ##
##). ## ## ## ## ## ResourceSpecifier #####. ## ## ##
##### # ##### ##### UIMAFramework.produceAnalysisEngine, #####
## ##### ## ##### ## AnalysisEngine #####.

```


 ##### ##### ## ##### ##### ##### ##### ##### ##### ## #####. ## ### ##
 ## ##### ##, ## ##### 3.2.5, ##### [83].

3.2.2. Analyzing Text Documents

#####. ### ## #####
 ### ## #####, ##### ## ##### ## ##### ## ##### 5, #####
 ##### ## ##### ## ##### ## ##### ## ##### ## ##### ## #####
 #####, ##### ## ##### ## ##### ## ##### 4, ##### ##
 #####. ##### ##, ##### ## ##### ##
 ##### ## #####; ## ##### 5, #####
 ##### [123] ## #####.

#####:

###

```
//create a JCas, given an Analysis Engine (ae)
JCas jcas = ae.newJCas();

//analyze a document
jcas.setDocumentText(doc1text);
ae.process(jcas);
doSomethingWithResults(jcas);
jcas.reset();

//analyze another document
jcas.setDocumentText(doc2text);
ae.process(jcas);
doSomethingWithResults(jcas);
jcas.reset();
...
//done
ae.destroy();
```

###

```
//create a CAS
CAS aCasView = ae.newCAS();

//analyze a document
aCasView.setDocumentText(doc1text);
ae.process(aCasView);
doSomethingWithResults(aCasView);
aCasView.reset();

//analyze another document
aCasView.setDocumentText(doc2text);
ae.process(aCasView);
```

```
doSomethingWithResults(aCasView);
aCasView.reset();
...
//done
ae.destroy();
```

#####, ### ##### ### ## ## ##### ##### #####. ####, ### ##### ### ##### #####
#####:

1. ### ## ##### ##### ##### ## ## ##.
2. ##### ## ##### #####, ##### ## ## ## ## #####
3. ## ##### ## ## ##### ## ## ## ## ## ## ## ## ## ## ##
4. ##### ## ##### ## ##### #####() ##### ## ##### ## ##### #####

3.2.3. Analyzing Non-Text Artifacts

#####. ### ####
setDocumentText #####, ### ##### ## ## ##
##. ### ##### 5, ~~~~~
~~~~~ [123] #####.

### 3.2.4. Accessing Analysis Results

##### (### #####) ##### ## ##### ## ##### ## ## ##, ##### ##  
## ## #####. ##### ##### ## ##### ## ## #####. ##### ## ##  
##### ##### ## ##### ## ## ##### ##### uima.tcas.Annotation ##### ##  
#### ## ##### ##### ## Annotation ## ## ##### ## #####. ### ## ##  
##### ##### ##### ##### #####.

##### ##### # ##### ## ##### ## ##### ##### #####, ### #####  
##### ## ##### ##### ## ## ##### ## ##.

#### 3.2.4.1. Accessing Analysis Results using the JCas

###:

- ##### 1.3.3, ##### ## ##### ## ##### ##### [27]
- ##### 5, ~~~~~ ## ~~~~~
- ### ##### ## org.apache.uima.jcas.JCas.

#### 3.2.4.2. Accessing Analysis Results using the CAS

###:

- ##### 4, ~~~~~ ## ~~~~~



- ##### org.apache.uima.examples.PrintAnnotations, #####  
examples\src.
- ##### org.apache.uima.cas org.apache.uima.cas.text  
#####.

### 3.2.5. Multi-threaded Applications

```
#####  
#####  
#####:
```

```
public class MyApplication {  
    private AnalysisEngine mAnalysisEngine;  
    private CAS mCAS;  
  
    public MyApplication() {  
        //get Resource Specifier from XML file  
        XMLInputSource in = new XMLInputSource("MyDescriptor.xml");  
        ResourceSpecifier specifier =  
            UIMAFramework.getXMLParser().parseResourceSpecifier(in);  
  
        //create Analysis Engine here  
        mAnalysisEngine = UIMAFramework.produceAnalysisEngine(specifier);  
        mCAS = mAnalysisEngine.newCAS();  
    }  
  
    // Assume some other part of your multi-threaded application could  
    // call "analyzeDocument" on different threads, asynchronously  
  
    public synchronized void analyzeDocument(String aDoc) {  
        //analyze a document  
        mCAS.setDocumentText(aDoc);  
        mAnalysisEngine.process();  
        doSomethingWithResults(mCAS);  
        mCAS.reset();  
    }  
    ...  
}
```

```
#####  
#####  
#####  
#####  
#####  
#####
```

```
#####  
#####
```

(### #####, ## ##### ## # #####), ##### ## ## ## ## ##  
## #####.

[illegible]

# ### ##### (### ##### ##) .### # #####, # ~~~~~#  
#####. ### ##### ##, # ~~~~~  
#####. # #####, #####  
#####. # #####, #####  
#####. #####:

```

public class MyApplication {
    private CasPool mCasPool;

    private AnalysisEngine mAnalysisEngine;

    public MyApplication()
    {
        //get Resource Specifier from XML file
        XMLInputSource in = new XMLInputSource("MyDescriptor.xml");
        ResourceSpecifier specifier =
            UIMAFramework.getXMLParser().parseResourceSpecifier(in);

        //Create multithreadable AE that will
        //Accept 3 simultaneous requests
        //The 3rd parameter specifies a timeout.
        //When the number of simultaneous requests exceeds 3,
        // additional requests will wait for other requests to finish.
        // This parameter determines the maximum number of milliseconds
        // that a new request should wait before throwing an
        // - a value of 0 will cause them to wait forever.
        mAnalysisEngine = UIMAFramework.produceAnalysisEngine(specifier,3,0);

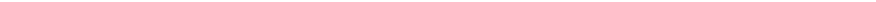
        //create CAS pool with 3 CAS instances
        mCasPool = new CasPool(3, mAnalysisEngine);
    }

    public void analyzeDocument(String aDoc) {
        //check out a CAS instance (argument 0 means no timeout)
        CAS cas = mCasPool.getCas(0);
        try {
            //analyze a document
            cas.setDocumentText(aDoc);
            mAnalysisEngine.process(cas);
            doSomethingWithResults(cas);
        } finally {
            //MAKE SURE we release the CAS instance
            mCasPool.releaseCas(cas);
        }
    }
}

```

##### ## ### ##### ##### ##### ##### ##### ##### #####. #####, #####  
## ### ##### ##### ## ### ##### #####, ##### ## #####  
## ##### ##### ## #####<sup>1</sup>. #####, ##### ## ##### # ##### ## ## ##  
#####. ## ### ##### # ##### ##### 3 #####. ## ### ##### #####, ##  
##### ## # ##, ## ##, ## ##### ##### ##.

#####. ##### (##### ## #####, ### ## ##### ## ##) #####  
## ## #####. ## ##### ## ##### ##### ## ##### ## ##### ## ##### #####  
##### ## ## # ## ##### ## ##### ##### ##### ##### ##### ##### ## # ## #####  
#####. ## ##### ##### ##### ## ##### ##### #####, ##### ## ## ## #####  
##### ##### ##### ##### ##### ## #####, ##### ## ## ##.

#####() ##### # ## ##### ## #####  
#####. ## ##### #####, ##### ## ##### 5,  
 [123].

#####  
#####  
#####  
#####

```
##### 0 ##### ## ### #####.#####() ##### ## # ##### #####. ## #### ## ###
## # ##### #####,## ## ### ##### ##### ## ##### ##### #####
#### ## ## ##### ## ##### ##### ## ## ####.## #### #####,### #####
##### ##### #####,### ## ##### ## ## #####,#### ## ##
#### ## ## ##### #####. # ##### ## 0 #### ##### ## ##### ## ## ##### ##,
##### #####.
```

### 3.2.6. Using Multiple Analysis Engines and Creating Shared CASEs

## #####, ## ##### ## ## ##### ##### ##### ##  
##### ## ## ##### ## ##### ##. ## #####, ## ##### 1.3,  
##### [22]. ## ## ## ## ##  
##### ## ## ## ## ## ## ## ##.

```
#####
##### # ##### #,### ## ## ## ## ## ## ## ## ## ## ## ##
## ## AnalysisEngine ##### ## ## ## ## ## ## ## ## ## ## ##
##### ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##
##### ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##:
```

1 #####  
#####  
#####  
#####.

```
AnalysisEngineDescription aeDesc1 =
    UIMAFramework.getXMLParser().parseAnalysisEngineDescription(...);

AnalysisEngineDescription aeDesc2 =
    UIMAFramework.getXMLParser().parseAnalysisEngineDescription(...);

CasConsumerDescription ccDesc =
    UIMAFramework.getXMLParser().parseCasConsumerDescription(...);

List list = new ArrayList();

list.add(aeDesc1);
list.add(aeDesc2);
list.add(ccDesc);

CAS cas = CasCreationUtils.createCas(list);

// (optional, if using the JCas interface)
JCas jcas = cas.getJCas();
```

### 3.2.7. Saving CASEs to file systems

### 3.3. Using Collection Processing Engines

UIMA Version 2.3.0

### 3.3.1. Running a Collection Processing Engine from a Descriptor

### 3.3.2. Configuring a Collection Processing Engine Descriptor Programmatically

```
//Creates descriptor with default settings
CpeDescription cpe = CpeDescriptorFactory.produceDescriptor();

//Add CollectionReader
cpe.addCollectionReader([descriptor]);

//Add CasInitializer (deprecated)
cpe.addCasInitializer(<cas initializer descriptor>);

// Provide the number of CASes the CPE will use
cpe.setCasPoolSize(2);

// Define and add Analysis Engine
CpeIntegratedCasProcessor personTitleProcessor =
    CpeDescriptorFactory.produceCasProcessor ("Person");

// Provide descriptor for the Analysis Engine
personTitleProcessor.setDescriptor([descriptor]);
```

```
//Continue, despite errors and skip bad Cas
personTitleProcessor.setActionOnMaxError("terminate");

//Increase amount of time in ms the CPE waits for response
//from this Analysis Engine
personTitleProcessor.setTimeout(100000);

//Add Analysis Engine to the descriptor
cpe.addCasProcessor(personTitleProcessor);

// Define and add CAS Consumer
CpeIntegratedCasProcessor consumerProcessor =
CpeDescriptorFactory.produceCasProcessor("Printer");
consumerProcessor.setDescriptor([descriptor]);

//Define batch size
consumerProcessor.setBatchSize(100);

//Terminate CPE on max errors
personTitleProcessor.setActionOnMaxError("terminate");

//Add CAS Consumer to the descriptor
cpe.addCasProcessor(consumerProcessor);

// Add Checkpoint file and define checkpoint frequency (ms)
cpe.setCheckpoint("[path]/checkpoint.dat", 3000);

// Plug in custom timer class used for timing events
cpe.setTimer("org.apache.uima.internal.util.JavaTimer");

// Define number of documents to process
cpe.setNumToProcess(1000);

// Dump the descriptor to the System.out
((CpeDescriptionImpl)cpe).toXML(System.out);
```

```
### ### ##### ### ### ##### ##### ##### ##:
```

```
<?xml version="1.0" encoding="UTF-8"?>
<cpeDescription xmlns="http://uima.apache.org/resourceSpecifier">
  <collectionReader>
    <collectionIterator>
      <descriptor>
        <include href="[descriptor]" />
      </descriptor>
      <configurationParameterSettings>...
    </configurationParameterSettings>
    </collectionIterator>

    <casInitializer>
      <descriptor>
        <include href="[descriptor]" />
      </descriptor>
    </casInitializer>
  </collectionReader>
</cpeDescription>
```

```

        </descriptor>
        <configurationParameterSettings>...
    </configurationParameterSettings>
</casInitializer>
</collectionReader>

<casProcessors casPoolSize="2" processingUnitThreadCount="1">
    <casProcessor deployment="integrated" name="Person">
        <descriptor>
            <include href="[descriptor]"/>
        </descriptor>
        <deploymentParameters/>
        <errorHandling>
            <errorRateThreshold action="terminate" value="100/1000"/>
            <maxConsecutiveRestarts action="terminate" value="30"/>
            <timeout max="100000"/>
        </errorHandling>
        <checkpoint batch="100" time="1000ms"/>
    </casProcessor>

    <casProcessor deployment="integrated" name="Printer">
        <descriptor>
            <include href="[descriptor]"/>
        </descriptor>
        <deploymentParameters/>
        <errorHandling>
            <errorRateThreshold action="terminate"
                value="100/1000"/>
            <maxConsecutiveRestarts action="terminate"
                value="30"/>
            <timeout max="100000" default="-1"/>
        </errorHandling>
        <checkpoint batch="100" time="1000ms"/>
    </casProcessor>
</casProcessors>

<cpeConfig>
    <numToProcess>1000</numToProcess>
    <deployAs>immediate</deployAs>
    <checkpoint file="[path]/checkpoint.dat" time="3000ms"/>
    <timerImpl>
        org.apache.uima.reference_impl.util.JavaTimer
    </timerImpl>
</cpeConfig>
</cpeDescription>

```

## 3.4. Setting Configuration Parameters

```

#####
##### (### ##### 1.2.1, ##### [15].
#####
#####

```

- ##### ##### ### ## ##### ## # #####, ## ##### ## ##### ##  
##### ## ##, ##
- ##### ## ##### ## ## ## #####.

```
##### ## ##### ##### ## ##### ## ##### ## ## ## ## ##
#####. ## ## ## ##### ## ## ## ## #####
##### ## ##### ##### ##### #####. ## #####, ## ##
##### # ##### ##### ## ##### UIMAFramework.getXMLParser().parse...
#####, ## ## ## ## ##### ##### #####() ##### ## ## ##
#####, ## ## ## ##### ##### ##### ## ##
### #####. ##### ## ##, ## ## ## #####
##### ## ## #####. ##### ## ##, ## # ## #####
#####:
```

```
// Create a description object by reading the XML for the descriptor

CasConsumerDescription casConsumerDesc =
    UIMAFramework.getXMLParser().parseCasConsumerDescription(new
        XMLInputSource("descriptors/cas_consumer/InlineXmlCasConsumer.xml"));

// get the settings from the metadata
ConfigurationParameterSettings consumerParamSettings =
    casConsumerDesc.getMetaData().getConfigurationParameterSettings();

// Set a parameter value
consumerParamSettings.setParameterValue(
    InlineXmlCasConsumer.PARAM_OUTPUTDIR,
    outputDir.getAbsolutePath());
```

```
#### ## ##### ##### #####:
```

```
CasConsumer component =
    UIMAFramework.produceCasConsumer(casConsumerDesc);
```

```
# #### ## ##### # ##### ## ##### ## ##### #####
##### ## ## ## #####. ## ## ## ## #####
##### ##, ##
```

```
component.setConfigParameterValue(
    "<parameter-name>",
    "<parameter-value>");
```

```
### #### ## ##### ## ##### ## ##### ## #####
#####:
```

```
component.reconfigure();
```

```
##### ##### ##### ## ## # ## #####, ## ##### ## ## ##
### ##### ## #####.
```





[illegible]

### 3.5.1.2. Building and Running a CPE including the Semantic Search CAS Indexer

---

UIMA Version 2.3.0 Application Developer's Guide 93

```

### ### ### ##### cpeGui ##### bin
##### (##### 2,
#####)

#####:
    • #####: %UIMA_HOME%/examples/descriptors/collectionReader/
      FileSystemCollectionReader.xml
    • #####: #####; #####/#####, #####
      #####
      #####
      • %UIMA_HOME%/examples/descriptors/analysis_engine/
        SimpleTokenAndSentenceAnnotator.xml
      • %UIMA_HOME%/examples/descriptors/tutorial/ex6/UIMAMeetingDetectorTAE.xml
    • #####:
      • %UIMA_HOME%/examples/descriptors/cas_consumer/SemanticSearchCasIndexer.xml
      • %UIMA_HOME%/examples/descriptors/cas_consumer/XmiWriterCasConsumer.xml

### ## #####:
    • ### ### ##### ##### ##### ##### ##### ## #####
      ### %UIMA_HOME%/examples/data #####.
    • ### ### ##### ##### ##### ##### #####
      ##### ## ##### ## %UIMA_HOME%/examples/descriptors/tutorial/search/
        MeetingIndexBuildSpec.xml
    • ### ### ##### ##### ##### ##### ##### ## #####
      ##### ##### ##### ##### ##### ## #####.

      #####: ### ##### ##### ##### ## ##### ##
        #####.
    • ### ### ##### ##### ##### ##### ##### ## #####
      ##### ##### ##### ## ##### ## ##### #####
      ##### ##### #####.

##### ## ### #####. ##### ## #####, # ##### ##### #####, ##
##### ## ### ##### ##### ##### ## ##### ## ##### ##### ## ###.

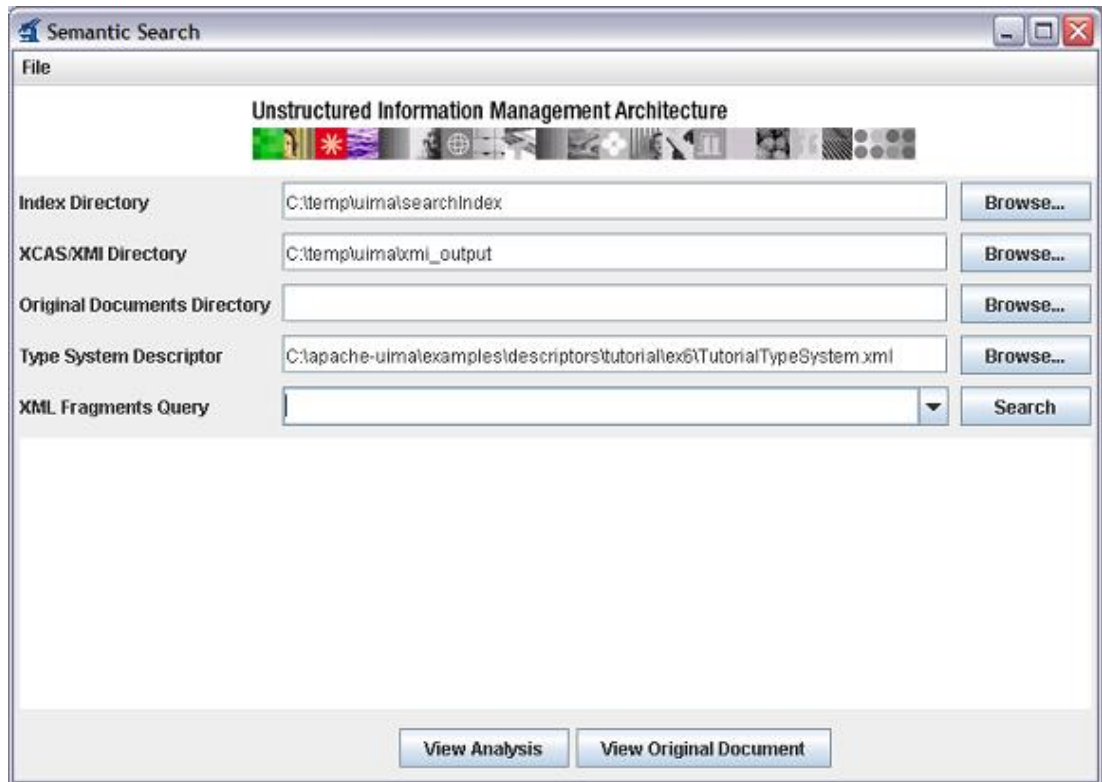
```

## 3.5.2. Semantic Search Query Tool

```

#####
#####
## ### #####, ### ## ##### ##### ## #####:
#####, ##### ## ## /bin ##### ## #####,
## ### #####. ## ### ## #####, ### ##### ## #####,
##### ## # ## ##### ## ## ## ##### ##, ##### UIMA
Semantic Search. #####:

```



##### ## ##### ## ##### ## #####:

- ##### ##### ##### ##### ##### ##### ##### ##### #####. ##### ## #####  
##### ##### ##### ##### ##### ##### ##### ##### ##### ##### #####  
##### ##### ## ##### #####
- ##### #####/##### ##### ##### ##### ##### ##### #####  
## ##### #####. ##### ## ##### ##### ##### ##### #####  
##### ##### ##### ## ##### ##### ##### ##### #####
- #####, ##### ##### ##### ##### ##### ##### #####  
##### ##### ##### ##### ##### ##### ##### #####. ##### ## #####  
##### ##### ##### ##### #####
- ##### ##### ##### ##### ##### ##### #####  
##### ##### #####. ##### #####, ##### ##### ## %UIMA\_HOME%/  
examples/ descriptors/tutorial/ex4/TutorialTypeSystem.xml

###, ## ##### #####, ##### ##### ## ##### #####  
##### ##### ##### ##### ##### ##### ##### ##### (#.#.  
<Meeting>UIMA</Meeting>). ##### ##### ##### #####  
##### ##### ##### ## ####://####.#####.###.###/####/####.

##### ## ##### # ##### ##### ##### #####, # ##### ## #####  
##### ## ## ##### ##### ##### ##### ##### ## ##### ## #####  
#####

##### ##### ## ##### ##### ##### ##### ## ## examples/src/com/ibm/  
apache-uima/search/examples/SemanticSearchGUI.java. #####

### 3.6. Working with Remote Services

### 3.6.1. Deploying a UIMA Component as a SOAP Service

96 Application Developer's Guide UIMA Version 2.3.0

```

• ### ### ##### ##### ##### ##### ##### ##### ##### ## ## uima-
  examples.jar ##### ## ##### ## ## %UIMA_HOME%/lib #####.

• ## #####, ## ## ##### ##### ## ##### ##### ##### ## ## #####,
  ##### ##### ##### ## ## ##### ## ## #####, ## ## ##### ## ##
  %CATALINA_HOME%/webapps/axis/WEB-INF/classes.

  ## ## #####, ## ## ## ##### ##### #####
  (##### ## examples/descriptors/analysis_engine/
  GovernmentOfficialRecognizer_RegEx_TAE) ## ## #####, ## ## ## ##
  ## ## ## examples/resources/GovernmentTitlePatterns.dat ##### ../WEB-
  INF/classes.

##### ##### ##### ## ##### ## ## ##### ## ##### ## http://
localhost:8080/axis/happyaxis.jsp ## #####. ##### ## ## #####
##### ## ## ## ## ##### ## ## ##### ## ##. ## ##### ## ##
## ## #####.###, ##### ## ## ## ## ##.###.###.

##### ##### ##### #####, ## ## ##### ##### ## ##
##### ## ## ## ##### ## ##### ## deploytool #####, ##### ## ##### ##
## /bin ##### ## ## ## ##. deploytool ## ## ##### ## ## #####
##### ## ## ##### ## ## ##### ##### ##### (#####); ##### ##
##### ## ## ##### ## ## examples/deploy/soap ##### ## ## ##. #####
##### ## ## ##### ## ## ##### ## ## ##### ## ## ##
##### ##### ## ## ## ## ##.

## ## #####, ## ## ## ## ## ##### ## ## ##### ## ## #####
##### ## (##### ## ## ## ## ##):

```

```

<deployment name="PersonTitleAnnotator"
  xmlns="http://xml.apache.org/axis/wsdd/"
  xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">

  <service name="urn:PersonTitleAnnotator" provider="java:RPC">

    <parameter name="scope" value="Request"/>

    <parameter name="className"
      value="org.apache.uima.reference_impl.analysis_engine
        .service.soap.AxisAnalysisEngineService_impl"/>

    <parameter name="allowedMethods" value="getMetaData process"/>
    <parameter name="allowedRoles" value="*" />
    <parameter name="resourceSpecifierPath"
      value="C:/Program Files/apache/uima/examples/
        descriptors/analysis_engine/PersonTitleAnnotator.xml" />

    <parameter name="numInstances" value="3"/>

    <!-- Type Mappings omitted from this document;

```

you will not need to edit them. -->

```
<typeMapping .../>
<typeMapping .../>
<typeMapping .../>

</service>

</deployment>
```

```
## ##### ##### ##### ## ##### ##### ## ##### ##### ## #####,####
##### ## ##### ##### ## ##### ##### (##### ##, ##### ##, ##
##### #####) ##### ##### ##### ## #####.

### numInstances ##### ## ##### ## ##### ##### ##
##### ## ## #####. ##### ## ##### ## ##### #####
#####. ##### # ## ##### ##, ## ## ## ##### ## ##, ## ##
##### ## ## ##### ## ##### #####.

## ##### ## ##### #####, ##### ## #####:
```

```
C:/Program Files/apache/uima/bin>deploytool
../examples/deploy/soap/Deploy_PersonTitleAnnotator.wsdd
```

```
#### ## ## ##### ## ##### ## # #####, ##### ## ##
##### ##### (#., http://localhost:8080/axis) ##
##### ## ## ##. ##### ## # ##### ## ##
#####, ##### ## ##### ## ## ##### ## #####.

### ##### ## ## #####
Deploy_PersonTitleAnnotator.wsdd #### ## ## ##### ##### ## ##
#####. ### ##### ## ## ##### ## ## ## ##
### #####.

####: ## deploytool ##### ## ## ## ## ##
##### ## http://localhost:8080/axis. ## ## ## ##, ## ##
#### ## ## ## #####.

#### ## ## ##### ## # ## ##, ## ## ## ## #
#####. ## ##### 3.6.3, ##### # ##### [100] ## #####.
```

### 3.6.2. Deploying a UIMA Component as a Vinci Service

```
##### ## ## ##### ##### ## ##### ## #####. ## #####
##### ## ## ## ##. #####, ##### ## ## ## ##### ## ##
## ##### ## ##### ##### (##), ## ##### ## ##### ##### 3.6.5, ###
##### ##### (##) [102].

## ##### # #####, ## ## ## ##### ##### ## ##
## ## ## ## ##. ## ## ## ## ## ## ## ##
```



```
#####
```

```
## ##### ##### ##### ##### ## ##### ##### ## ##### ##### ## ##
##### , ##### ##### ## ##### ##### ## ##### (##### ## , #####
##### , ## ##### #####) ##### ##### ##### ## ## #####.
```

```
### serverSocketTimeout ##### ### ##### ## ##### (##### = 5
#####) ##### ##### ##### ##### ##### #####. #####
##### ## ####, ### ##### ##### ##### ##### ##### ##### # ## ## #####
###, ##### ## ##### ## ## #####. ### ##### ## ##### ## ## #####
##### ## # ##### ##### ##### ##### ## ##### ##### ##### #####
##### (#####).
```

- `<parameter name="threadPoolMinSize" value="[Integer]"/>`: #####  
#####  
#####  
#####.

---

|                    |                               |    |
|--------------------|-------------------------------|----|
| UIMA Version 2.3.0 | Application Developer's Guide | 99 |
|--------------------|-------------------------------|----|

```
##### threadPoolMinSize,#####
##### threadPoolMaxSize ##

### startVinciService #####
#####_#####
#####.#####
#####)#####
#####
#####;
#####
#####.

#### #####,#### #
#####.### 3.6.3,##### \[100\] #####
```

### 3.6.3. How to Call a UIMA Service

```
#### # #####
#### # #####
##### #
##### #
##### #
```

```
## ## ##,### ## # #####
##### #
##### #
##### #
##### #
examples/descriptors/soapService ### examples/descriptors/vinciService.###
##### #
```

```
####,##### #
## #####.## ## ## bin
#####
##### ## CATALINA_HOME
### ##### (### ##)
```

- #####.###
- #####.###
- #####.###
- #####.###
- #####.###
- #####.###

#### 3.6.3.1. SOAP Service Client Descriptor

```
### #####
##### #:
```

```
<uriSpecifier xmlns="http://uima.apache.org/resourceSpecifier">
```

```
<resourceType>AnalysisEngine</resourceType>
<uri>http://localhost:8080/axis/services/urn:PersonTitleAnnotator</uri>
  <protocol>SOAP</protocol>
  <timeout>60000</timeout>
</uriSpecifier>
```

```
### <#####> ##### ### ##### ##### ##### ## #####. ###
##### ### ## ##### ## ##### ## ## ## ##### #####.

### <###> ##### ##### ##### ## ####. ## ##### ## ## (#####,
## #### #####) ## ## ##### ## (###:#####), ##### ## ##
#### ##### ## ## ##### ##### ## ##### ## #####.
```

### 3.6.3.2. Vinci Service Client Descriptor

## ##### # ##### #####, # ##### ##### ## ####:

```
<uriSpecifier xmlns="http://uima.apache.org/resourceSpecifier">
  <resourceType>AnalysisEngine</resourceType>
  <uri>uima.annot.PersonTitleAnnotator</uri>
  <protocol>Vinci</protocol>
  <timeout>60000</timeout>
  <parameters>
    <parameter name="VNS_HOST" value="some.internet.ip.name-or-address"/>
    <parameter name="VNS_PORT" value="9000"/>
  </parameters>
</uriSpecifier>
```

```
#####  
#####  
(uima.annot.PersonTitleAnnotator)#####  
#####
```

```
### ### ##/## ### ##### ### ##### ##### (###) ##### ## ##### ##
## ##### ## ## ##### <#####> #####. ## ## #####, ## ##
## ##### ## ## ##### ##### ## ## ##### ## (##) #####) ##### -
DVNS_HOST=<host> ### -DVNS_PORT=<port> ##### #####. ## ## ##### ## ##
##### ##, ##### ## ##: ##### ## ## VNS_HOST, ## 9000 ## ##
VNS_PORT. ## ## ##### ## ##### ## ##### ## # ## #####.
```

### 3.6.4. Restrictions on remotely deployed services

#####  
#####  
#####  
#####  
#####  
#####  
#####

```

### ##### #####, ### ## ##### # ## ## ## ##, ##### ## ##
### #####, ##### ##### ## ## ## ## ## ## ##.
##### ## ## ##### ## ## ## ## ## ## ## ## ## ## ##
##### ## ## ## ##, ## ## ## ## ## ## ## ## ## ##
### ## ##, ##### ## ## ##### ## ## ## ## ## ## ## ##
## ## ##. ##### ## ## ##, # ##### ## ## ## ## ## ##
##### ## ## ## ## ## ##.

```

```

####: ### ##### ## ## ## ##, ## ## ##
##### ## / ## ## ## ## ## ## ##.

```

### 3.6.5. The Vinci Naming Services (VNS)

```

##### ## ##### ## #####, ##### ##
##### ## ##, ## ## ##### ## ## ## ##.
## ##### ## ## ##, ##### ## ## (## ##
#####).

```

```

## ##, ##### ## ## ## ## ## ## ## ## ## ## ##
## ## ## ##. ##### ## ## ## ## ## ##
## ## ##, ## ## ## ## ##. ## ## ## ## ##
## ## ## ## ## ## ## ## ## ## ## ## ## ## ##
## ## ##. ## ## ## ## ## ## ## ## ## ##
##.#####.

```

```

## ##, ##### ## ## ## ## ##. ## ## ## ## ##
##### ## ## ## ##, ## ## ## ## ## ##
##### ## ## ##, ##### ## ## ## ## ##
#####. ## ## ## ## ## ## ## ## ## ##
##### ## ## ## ## ## ## ## ##.

```

#### 3.6.5.1. Starting VNS

```

## ## ## ## ## startVNS ##### ## ## bin ##### ## ##
#####, ## ## ## ##. ## ## ## ## uimaj-examples
#####, ## ## ## ## ## ## ## ## ## ## ##
##### ## → ## → ##... ## ## ## ##.

```

```

####: ## ## ## 9000 ## ## ## ## ## ## ##
#####. ## ## ## ## ## ##:

```

```

java.net.BindException: Address already in use:
JVM_Bind

```

```

## ##### ## ## ## ## 9000. ## ## ##, ##
## ## ## -p <port> ## ## startVNS ## ##, ## <port> ## ## ##
##### ## ##.

```

```
[10/6/04 3:44 PM | main] WARNING: Config file doesn't exist,  
    creating a new empty config file!  
[10/6/04 3:44 PM | main] Loading config file : .vns.services  
[10/6/04 3:44 PM | main] Loading workspaces file : .vns.workspaces  
[10/6/04 3:44 PM | main] =====  
(WARNING) Unexpected exception:  
java.io.FileNotFoundException: .vns.workspaces (The system cannot find  
the file specified)  
    at java.io.FileInputStream.open(Native Method)  
    at java.io.FileInputStream.<init>(Unknown Source)  
    at java.io.FileInputStream.<init>(Unknown Source)  
    at java.io.FileReader.<init>(Unknown Source)  
    at org.apache.vinci.transport.vns.service.VNS.loadWorkspaces(VNS.java:339)  
    at org.apache.vinci.transport.vns.service.VNS.startServing(VNS.java:237)  
    at org.apache.vinci.transport.vns.service.VNS.main(VNS.java:179)  
[10/6/04 3:44 PM | main] WARNING: failed to load workspace.  
[10/6/04 3:44 PM | main] VNS Workspace : null  
[10/6/04 3:44 PM | main] Loading counter file : .vns.counter  
[10/6/04 3:44 PM | main] Could not load the counter file : .vns.counter  
[10/6/04 3:44 PM | main] Starting backup thread,  
    using files .vns.services.bak  
and .vns.services  
[10/6/04 3:44 PM | main] Serving on port : 9000  
[10/6/04 3:44 PM | Thread-0] Backup thread started  
[10/6/04 3:44 PM | Thread-0] Saving to config file : .vns.services.bak  
>>>>>>>>>> VNS is up and running! <<<<<<<<<<<<<<<<  
>>>>>>>>>> Type 'quit' and hit ENTER to terminate VNS <<<<<<<<<<<<<<<<  
[10/6/04 3:44 PM | Thread-0] Config save required 10 millis.  
[10/6/04 3:44 PM | Thread-0] Saving to config file : .vns.services  
[10/6/04 3:44 PM | Thread-0] Config save required 10 millis.  
[10/6/04 3:44 PM | Thread-0] Saving counter file : .vns.counter
```

- `vns.services`
- `vns.counter`



### 3.6.6. Configuring Timeout Settings

```
#####  
#####  
#####  
#####
```

- [illegible]

## #####  
## ~~~~~  
#####.  
#####.

##### ## ##### ##### ## ####, ## ~~~~~ ##  
 ~~~~~. ### ##### ## ##### ## ##  
 #####, ## ##### ## ##### ## ##### ## ##### ##
 ##### ## ##. ## ##### ## ## ##### ## ##### ## ##. ##
 ##### ##### (#####) ##### ## ##### ##### ## #####
 ##### ##### ## #####. ##### ## ##### ##, ## #####
 ## ##### ## ## ## ## ## ##, ##### ## ##
 #####. ## ## ## ## ## ## ## ## ## ## ## ##
 ## ##### (#####).

3.6.6.1. Setting the Client Timeout

 ## ## ##### ## (## ##).

```
## ## ## ##### ## ##### ##### ##### ##### ## ## ## ##, ## ## ## ##
## ##### # ## ## ##, ##### ## ##### ##### ## ##### ## ## #####
##### (## ##### 3.6.3, ##### # ## ##### [100]). ## #####:
```

```
<uriSpecifier xmlns="http://uima.apache.org/resourceSpecifier">
  <resourceType>AnalysisEngine</resourceType>
  <uri>uima.annot.PersonTitleAnnotator</uri>
  <protocol>Vinci</protocol>
  <timeout>60000</timeout>
  <parameters>
    <parameter name="VNS_HOST" value="some.internet.ip.name-or-address"/>
    <parameter name="VNS_PORT" value="9000"/>
  </parameters>
</uriSpecifier>
```

60000. ### ##### ##

#####. ##
#####. ## ##### ## ## ## ## ##.

```
## ### ## ##### ## ##### ## ##### ##### ## ## ##, ###
## ##### ##### ## ##### ## ##### ##### errorHandling #####.##
#####:
```

```
## ## ### #####,### ##### ## ## ## 60000,### ##### ##
##### ## ##### ## ## ## ## ## ## ## ## ## ## ##
#####.
```

```
##### GetMetaData #####
#####. #####
##### (### 60 #####). #####, ## #####
##### ## #####, ## ## #####
##### 2.7, #####
```

```
### ##### ##### ##### ##### ## ##### #####, ## ## ##### ## ## #####
##### ##### ## ##### ## ##### ##### 3.6.2, ##### ## # #####
##### [98]. ## #####:
```

106


```

<parameter name="numInstances" value="1"/>

<parameter name="serverSocketTimeout" value="120000"/>

</service>

</deployment>

```

```

### ##### ##### ##### ## ## ## 120000 #####, ## ## #####. ###
##### ##### ## ## ## ##### ##### ##### ##### ## #####
#####. ##### ## ## ## ##, ## ##### ## ##### ## ##### ## ##
##### # ## ## ##### ##, ##### ## ##### ## ## #####. ## ## ## ##
##### ## ## ##### ## ## ## ## ## ## ## ## ##### ## ##
##### (#####). ## ##### ## ## ## ## ## ## ## ## ##
### ##### ## ## ##### ##### #####.

## ## ##, ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##. ## ## ##
#####. #####, ## ## ## ## ## ## ## ## ## ## ## ## ## ##
#####, ##### ## ## ## #####, ## ## ## ## ## ## ## ## ##
##### ##### ##### ## ## ##. ## ## ## ## ## ## ## ## ##
##### ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##
### ##### ##### #####.

```

3.7. Increasing performance using parallelism

```

##### ## ##### ## ## ## ##### ##### ## ##### ##### ## ## ##
#####. ##### ## ## ##### ## ## ##### ##### ## ## ##
##### ## ## ## (##### ## ## #####) ##
##### ##### ## ## ## ## #####.

### ##### ##### ## ## ##### ## ## ## ## ## ## ## ## ##
#### ## #####. #### ##### ##### ##### ## ## ##
##### ## ##, ## ## ## ## ## ## #####. ## ##### ##, ## ##
<casProcessors> #####, ## ## ##### processingUnitThreadCount,
##### ## ## ## ## ##### ## ## ## ## ## ## ## ## ##
#### 1, ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##
(## ##### ## <casProcessors> ## ## ## casPoolSize). ## ## ## ## ##
#####, ## ##### 3.6, ## ##### ## ~~~~~.

### ##### ##### ##### ##### ## ## #####
#####, ##### ## ##### ## ##, ##### ## ##### ## ##
##### #####. ## ##### ## ## ## ## ## ## ## ##, ## #####
## ##### ## ##, ## ##### ## ## ## ## ## ##, ## ## ## ##
##### ## #####.

##### ## ## #####: ## #####, ## ## #####. ##
##### ## ## ##### ## ## ## ## ## ## ## ## ## ##
#####. ## ## ## ##### ## ## ## ## ## ## ## ## ## ##

```

```
### ##### ## ##### #####; ### ##### ##### ## ##### ##### 2 ## ####
##### ## ## ##### #####.
```

```
### ##### ##### ##### # ##### ##### ##### ## ##### #####
##### #####. ##### ## ##### ## ##### # ##### ##### ## ##
<#####> ##### ## ## ## #####:
```

```
<#####>
<#####-#####-#####-#####> />
</#####>
```

```
## #### ## ## #####, ### ##### ##### ## ####.
```

```
## #####, ##### ##### ##### ##### ## ##### ##### # ##### #####
##### ## ##### ## ##### ## ##### ##### ##### (### ## <parameter
name="numInstances"> ## ##### ## ##### ##### ##### ##### 3.6,
##### ##### ##### \[96\] ##### ##### ## ##### ## #####
##### ## ## ##### ##### ## ## ##### (##### ## ##### ##
##### ##### # ##### ## ##### ## ##### ##### ## #####
#####).).
```

```
####: #### ##### ## ##### ##### (### ##### #####
##### ##### 3.6.1.5, <#####> ##### ## 
), #### ## ##### ## ####. ## ##### 
## # ##### ## #####, ## ##### ##### ## ## #####, ##### ##
##### ##### (### ##### 3.6.2, ##### ## # ##### \[98\]).
```

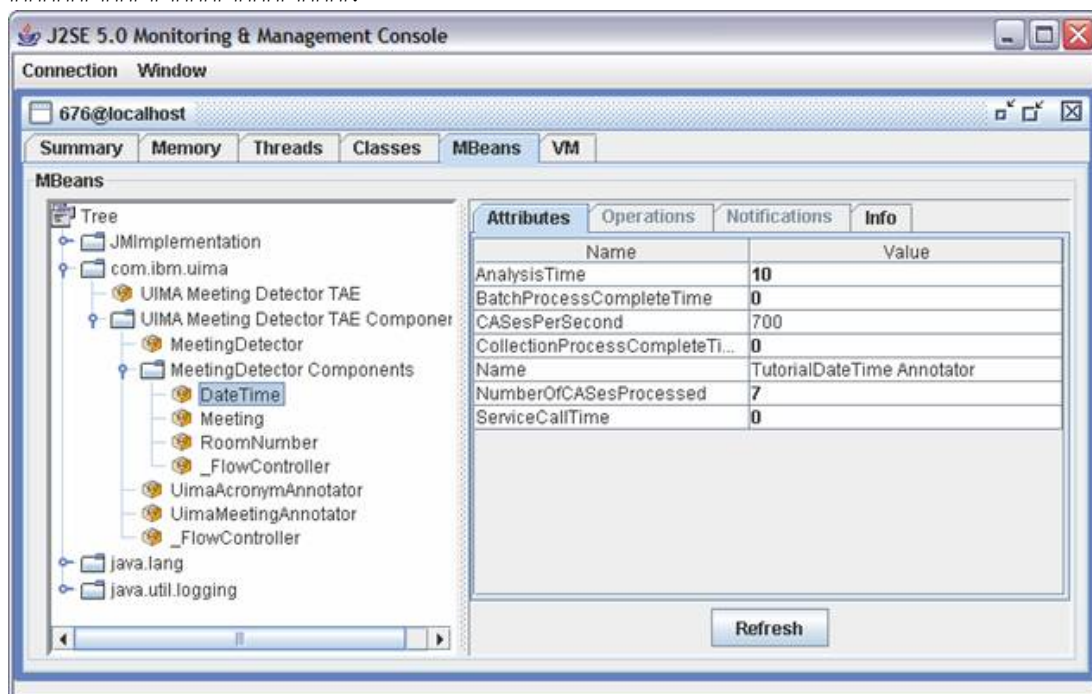
3.8. Monitoring AE Performance using JMX

```
## ## ##### 2, ##### ##### ##### ##### ## ##### ##### #####
### ##### ##### (###) ###. ### ## # ##### ##### ## ## #####
##### #5.0; ##### ## ##### # ##### ##### ##### ##### ## ##
1.4. ## ##### ## ## ## ##### ## ## ##: ####://####.###.###/#####/
#####/#2##/###.####. #### ## ## # ##### ## ## ## ##,
### ##### ##### ##### ##### ## ## ## ## #####
#####.
```

```
####: ### ## ## ##### ##### #####; ### ##### ## ## #####
##### ## ##### ##### (#### ## ## ## ##) ## #####
# ##### ## ## ##, #.#. -Dcom.sun.management.jmxremote.port=1098
-Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.ssl=false
```

```
###, ## ## ## ## ## ## ##### ## ## ## #####. ### 5.0 ## ##### #
##### ##### ## ## ## ##. ##### ## # ##### ##, ##### ## ## bin
##### ## ## ## ##, ## ##### ## jconsole #####. ##### ## ##
##### ##### ## ## ##### ## ## ## ##### #####
```

```
#####, ## ## ##### # ##### (## #####) ##### ## ###, #.#. #####:1098. ## ##
##### ## ## ## # ##### ## ##### ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##
##. ##### ## ## ##### ##, ##### #####.##### ## ## ## ## ## ## ## ## ## ##
##### ## # ##### #####:
#####
```



```
#### ## ### ##### ##### #org.apache.uima# ## ### ##### ##### ## ## ## ###
##### ##### ## ### ##### ##### ## ##### ##. ### ## ##### ## ## ##
##### ##### ## ##### ##### ##### ##### ## ## ##### ## ## #####.
```

#####, ##### ## ## ##### ## #####

#####, ## #####. ##### #### ## ## ##### ##### ####, ### ## ####. #### ##, ## ##

#####.

```
### AnalysisTime, BatchProcessCompleteTime, ### CollectionProcessCompleteTime
##### ### ### ##### ####, ## #####, #### ## ###
##### ## ### ##### process(), batchProcessComplete(), ###
collectionProcessComplete() #####, #####. (#### ### ## ### #####,
#### ##### ## ### hasNext() ### next() ##### ## #### ##### ###
#####.)
```


#####

```
#####  
#####
```

```
#####.#### ## #### ##### ## AnalysisEngine.PARAM_MBEAN_NAME_PREFIX
##### ##### ##### ##### #####:
```

```
//set up Map with custom JMX MBean name prefix
Map paramMap = new HashMap();
paramMap.put(AnalysisEngine.PARAM_MBEAN_NAME_PREFIX,
            "org.myorg:category=MyApp");

// create Analysis Engine
AnalysisEngine ae =
    UIMAFramework.produceAnalysisEngine(specifier, paramMap);
```

```
#####,### ## ## ## AnalysisEngine.PARAM_MBEAN_SERVER ##### ## #####
# ##### ##### ## # ## ##### ##### ##### ##### ##### #####
## ## ##### ##### ## ## ##### ##### ## ##### ##### (#### 5+
####).
```

```
#### ##### ## ## ## ## ##### ## ## #### 5 #####2.
```

3.9. Performance Tuning Options

```
##### ## # ##### ##### ## ##### ##### ##### ##### ## #####
### ##### ##### ## ##### #####. ##### ##### ##### ## ##
### ##### ##### ## ##### ##### ## #####. ### ##### # ####
##### ##### ##### ## ##### ##### ## ##### ## ## ## #####
#### ## ##### ## ##### #####. ##### ## ## #####.
```

```
XMLParser parser = UIMAFramework.getXMLParser();
ResourceSpecifier spec = parser.parseResourceSpecifier(
    new XMLInputSource(descriptorFile));
// Create a new properties object to hold the settings.
Properties performanceTuningSettings = new Properties();
// Set the initial CAS heap size.
performanceTuningSettings.setProperty(
    UIMAFramework.CAS_INITIAL_HEAP_SIZE,
    "1000000");
// Disable JCas cache.
performanceTuningSettings.setProperty(
    UIMAFramework.JCAS_CACHE_ENABLED,
    "false");
// Create a wrapper properties object that can
// be passed to the framework.
Properties additionalParams = new Properties();
// Set the performance tuning properties as value to
// the appropriate parameter.
additionalParams.put(
    Resource.PARAM_PERFORMANCE_TUNING_SETTINGS,
```

² ####://####.###.###/#2##/1.5.0/####/###/#####/#####/#####.#####_#####

```
        performanceTuningSettings);  
    // Create the analysis engine with the parameters.  
    // The second, unused argument here is a custom  
    // resource manager.  
    this.ae = UIMAFramework.produceAnalysisEngine(  
        spec, null, additionalParams);
```

```
#####
```

- UIMAFramework.JCAS_CACHE_ENABLED: #####

(#####). ####

#####
- UIMAFramework.CAS_INITIAL_HEAP_SIZE: #####

(#####). #### 32####

#####/

#####
- UIMAFramework.PROCESS_TRACE_ENABLED: #####
(#####). ####

org.apache.uima.util.ProcessTrace.
- UIMAFramework.SOCKET_KEEPALIVE_ENABLED: #####
(#####)

true.


```

##### next() #####
#####
##### # FinalStep.

### # FlowControllerContext,#####
UimaContext.##
### UimaContext,### FlowControllerContext #####
### #. #####
#####. ###
# FlowControllerContext ## getContext() #####
## JCasFlowController_ImplBase ## CasFlowController_ImplBase.###,###
FlowControllerContext.getAnalysisEngineMetaDataMap #####
# #####. ###
#####
#####,### AnalysisEngineMetaData #####.

##### addAnalysisEngines ##
removeAnalysisEngines.#####
## #,##
#####. #####
#####/
#####. #####
#####.

```

4.1.2. Example Code

```

#####
#####. ##
#####
#####.

###
org.apache.uima.examples.flow.WhiteboardFlowController ###
##### examples/src #####.

```

4.1.2.1. The WhiteboardFlowController Class

```

public class WhiteboardFlowController
    extends CasFlowController_ImplBase {
    public Flow computeFlow(CAS aCAS)
        throws AnalysisEngineProcessException {
        WhiteboardFlow flow = new WhiteboardFlow();
        // As of release 2.3.0, the following is not needed,
        // because the framework does this automatically
        // flow.setCas(aCAS);

        return flow;
    }
}

```



```

    }

    class WhiteboardFlow extends CasFlow_ImplBase {
        // Discussed Later
    }
}

```

```

### WhiteboardFlowController ##### CasFlowController_ImplBase ###
##### computeFlow #####. ##### computeFlow #####
#####; ## ##### WhiteboardFlow #####
#####. #####
#####.

##### WhiteboardFlow ###, ##
#####.

```

4.1.2.2. The WhiteboardFlow Class

```

class WhiteboardFlow extends CasFlow_ImplBase {
    private Set mAlreadyCalled = new HashSet();

    public Step next() throws AnalysisEngineProcessException {
        // Get the CAS that this Flow object is responsible for routing.
        // Each Flow instance is responsible for a single CAS.
        CAS cas = getCas();

        // iterate over available AEs
        Iterator aeIter = getContext().getAnalysisEngineMetaDataMap().
            entrySet().iterator();
        while (aeIter.hasNext()) {
            Map.Entry entry = (Map.Entry) aeIter.next();
            // skip AEs that were already called on this CAS
            String aeKey = (String) entry.getKey();
            if (!mAlreadyCalled.contains(aeKey)) {
                // check for satisfied input capabilities
                //(i.e. the CAS contains at least one instance
                // of each required input
                AnalysisEngineMetaData md =
                    (AnalysisEngineMetaData) entry.getValue();
                Capability[] caps = md.getCapabilities();
                boolean satisfied = true;
                for (int i = 0; i < caps.length; i++) {
                    satisfied = inputsSatisfied(caps[i].getInputs(), cas);
                    if (satisfied)
                        break;
                }
                if (satisfied) {
                    mAlreadyCalled.add(aeKey);
                    if (mLogger.isLoggable(Level.FINEST)) {
                        getContext().getLogger().log(Level.FINEST,
                            "Next AE is: " + aeKey);
                    }
                }
            }
        }
    }
}

```

```

        return new SimpleStep(aeKey);
    }
}
// no appropriate AEs to call - end of flow
getContext().getLogger().log(Level.FINEST, "Flow Complete.");
return new FinalStep();
}

private boolean inputsSatisfied(TypeOrFeature[] aInputs, CAS aCAS) {
    //implementation detail; see the actual source code
}
}

```

```

#### ##### ## ## WhiteboardFlowController ## ##### ## ##### # #####
###.# ##### ## ## ## ##### ## ##### ## ##### ## getCas() #####, ##### ##
# ##### ## ##### ## ## CasFlow_ImplBase #####.

```

```

#### ##### ## next ##### ## #####, ## ##### ##### ##### ## #####
## ## ## ##### ##### ##### (##### ## ## ## ## ## getContext().
getAnalysisEngineMetaDataMap) ## ## ## ## ## ##### ##
## ##### ##### ## ## ##### ## ## ## ## (## ##, ## ##
##### ## ##### ## ##### ## ## ##### ##### ## ## ##. ## #####
## ##### ## ##### ## ##### ## ## ## ## ## ## ##### ## ## ##
#####.##### ## ## ## ## #####.

```

```

#### ## ## ## ##### ##### ##### ##### ## ##### ##, ## #####
#### ## ##### # ##### ##### ## ## ## ## ## ##### ##### ##
##### ##:

```

```
return new SimpleStep(aeKey);
```

```

### ##### ##### # ##### ## ##### ##### ##### ## ## ##### ## ##
mAlreadyCalled #####, ## ##### ##### ## ## ##### #####. ##### ##
## # ##### #####. ## ## ##### ## ##### # ##### ##### ##
##### ##### ##### ## ## ##. #####, ## ## ## ## ## ## ## ## ##
#### ## ##### #####.

```

```

## ##### ## ## ##### ##### ## ## ## ## ##### ##### ##, ## ##
##### ##### ## ## ## ## ## ## ##### # ##### #####:

```

```
return new FinalStep();
```

```

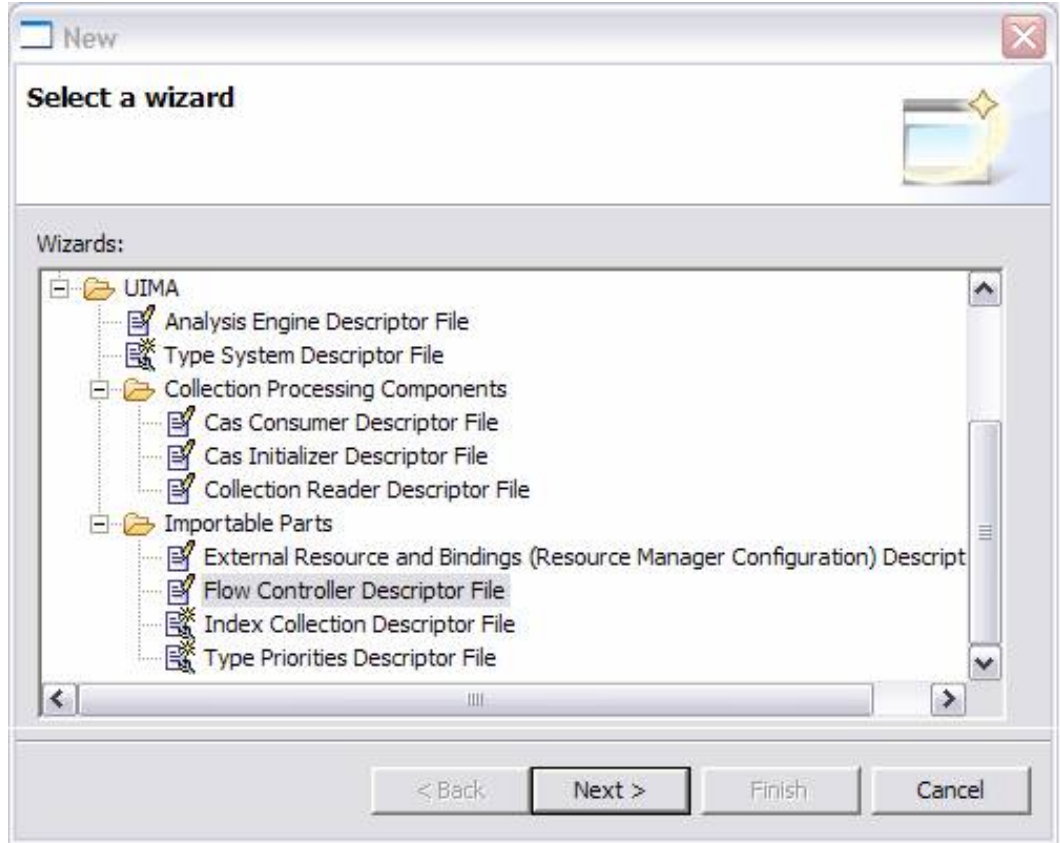
####, #### ## ## ## ## ## ## ## ## ## ##### ##### ##### ##
#### ## ## ## ## #####. #### ## # ## ## ##### ## ## ## ## ##
#### ##### ## ##### ## ## ##### ##.

```

4.2. Creating the Flow Controller Descriptor

```
## ##### # #### ##### ##### ## ### ##, ### ##### → ### → ##### → ##### →
##### ##### #####:

```



```
#### ##### ## ### ##### ##### ## ### ##### #####:

```


▼ Component Engine Flow

Choose a flow type and describe the execution order of your engines.
The table shows the delegates using their key names.

Flow Kind:

Flow Controller:

Key Name:

 aeconfiguration3
 aeconfiguration8
 aeconfiguration32
 aeconfiguration3

```
### ### ##### ## ##### ##### ##### ##### ##### ##### ##### #####
##### ##### #####. ## ### ##### ## ##### ##### #####, ### ### ## ## ## ##### ##
### ##### ##### ##### ##### #####, ### ##### ## ##### ## ##### ##
```

```
## ### ##### ##### ##### ##### ##### ##### ##### ## #####, ### ##### ## ##### #
##### ##### ##:
```

```
<delegateAnalysisEngineSpecifiers>
...
</delegateAnalysisEngineSpecifiers>
<flowController key="[String]">
  <import ../>
</flowController>
```

```
## #####, ### ### ## ##### ## ##### ## ##### ## ##### ## ##### # ## ##### 2.2,
##### ## ~~~~~.
```

```
### ### ##### ## ### ##### ##### ## ## ##### ##### ## ## #####
##### ##### ##### # ## ##### #####, #####, ### ##
#####.
```

4.4. Adding a Flow Controller to a Collection Processing Engine

```
##### ##### ## ##### ##### ## ##### ##### #####. ## ##
# ##### ## # ## ## ##### ## ##### ## ##### ## #####
##### ##### ##### ## ##### ##### #####, ### #####
##### ##### ## ##### ##. ### ##### ##### ##### ##### ## ##
#### # ##### ## ## ##### ##### ##### ## # ##.
```

4.5. Using Flow Controllers with CAS Multipliers

```
## ### ##### ##### ##### ##### ## ##### ## ##### #####
##### ##### # ## ##### (### ##### 7, ~~~~~)
```



```
## ### ##### ##### ##### ##### # #####, ## ## ##### ## ## #####
##### ## # #####,#### continueOnFailure ##### ## ##### ## ##
#####. ## #### ##### #####, ## ## ## ## ##### ##### ## ##
##### ##### ## # #####,#### ## continueOnFailure ##### ## ##
##### ## ##### ## ## #####. #### ##### ##### ##### #####
## ##### ## ## #####.
```

```
#### ## ## ## ##### ## ##### ## # ## ## ## ##### ##### #####
#####. #### ##### ## ## ##### ## ## ##### ## ##### #####
## ## ## ##### ## #####, ##### ## ## ##### #####.
```

```
## ### ####, ## ##### ## ##### ## ## ##### ## ## #####, #####
continueOnFailure ##### ##, ## ##### ## ## ## Flow.aborted()
##### ## ##### ## ## ##### ## ##### ## ## #####.
```

```
#### ## ##### ## ## ## ##### ## ## #####, ## ## #####
org.apache.uima.examples.flow.AdvancedFixedFlowController, ## ## examples/
src ##### ## ## ## ##. #### ##### ##### ##### ## ## ## ParallelStep.
```

Chapter 5. Annotations, Artifacts, and Sofas

```
## ## #####, ### ##### ## ##### ## ##### ## #####,
##### ## ##### ## ##### ## ##### ## #####. ###
##### ## ##### ## ##### ## ##### ## ##### ## #####,
##### ## ##### ## #####, ## ## ## ## ##### ## ## ##
## ##### ## #####.
```

5.1. Terminology

5.1.1. Artifact

```
### ##### ## ## ##### ## ## #####. ## ##### ##
## ##### ## ##, ## ##, # #####, # ##### ## ## ##, ## #####4
#####, ###. ##### ## ## ##### ## ## ## ## ## #####
##### ## ##. ## ##, ## ## ## ## ## ## ## ## ##
#####. ##### ## ##### ## ## ## ## ## ##
##### ## ## #####.
```

5.1.2. Subject of Analysis — Sofa

```
#### ##### ## ## ##### ## ##### # ##### ## #####, #####
#### ## ##### ##### ##### ## ##### ## #####. #####,
#### ## ##### ##### ## ##### ## ## ##### ## #####,
## ##### ##### # #####. ## #####, ## ##### ## ##
##### ## #####, ## ##### ## ##, ##### ##### #####
## ## ##### ## ## #####.
```

```
##### ## ##### ## #####, ##### ## #####: ## ##
## ##, # ##### ## ##, ## ##### ## ## ##, ## ## ##
#####, ##### ## ## # #####, ##.
```

```
#####, ##### ## ## ##### ##### ## # ##. ## ## ##### ## ## ##
##### ##### ##### ##### ## ## ##### ## ## ##, ##
## ## ##.
```

5.2. Formats of Sofa Data

```
#### ## ## ## ## #####, ##### ##### ## #####, ## #
## ##### ##### ## ## ##### ## # #####.
```

```
### ##### ## ##### ## ## ## ## ## ## ## ##, ## ##
##### ## ## ## ## ## ## ## ##, ##### ##, ##.
```

5.3. Setting and Accessing Sofa Data

- ```

● ## ### ##### ## #####, ### ##### ##### # #####. ### #####
#####.

 ● ## ### ##### ## ## ##### ##### ## #####, ##
 ##### ## ##### ## ##### ## ##### ## ##8 #####.

 ● ## ### ##### ## ## ## # #####, ### ##### ## ## ##### ##
 #####, ##### #####.

● ## ### ##### ## ## ##### ## # ##, ### ##### ##### ## #####
##.#####(). ### ##### ##### ##### ## #####
#####:##, #####:##, #####:## ## ## ## #####,
URLStreamHandlerFactory, ### ##### ## ## #####
##. ## ## ## ## ## ##://##.##.##/#2##/1.5.0/###/###/###/###/
#####.##.

```

## 5.4. The Sofa Feature Structure

### 5.5.1. Built-in Annotation types

### 5.5.2. Annotations have an associated Sofa

## 5.6. AnnotationBase

UIMA Version 2.3.0

[illegible]



---

## Chapter 6. Multiple CAS Views of an Artifact

#####  
#####  
#####

#####  
#####  
#####  
#####  
#####  
#####

#####  
#####  
#####  
#####  
#####  
#####

#####  
#####  
#####

#####  
#####  
#####

```
aView.addFsToIndexes(aFeatureStructure)
aView.removeFsFromIndexes(aFeatureStructure)
```

#####  
#####

---

### 6.1. CAS Views and Sofas

#####  
#####  
#####

---

#### 6.1.1. Naming CAS Views and Sofas

#####  
#####





```
#####:
#####
```


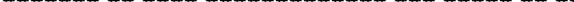


- ##### ### #####, ### ### ####, ##### ## ### ##### #####
- ##### # ##### ## ## ##### #### ## ### ##### #####, ## ####
- ##### # #### ## ##### ## ##### # ##### ##### ##### #####

### 6.2.3. Component XML metadata

```


#####,
#####:
```

```
<capabilities>
 <capability>
 <inputs/>
 <outputs/>
 <inputSofas>
 <sofaName>rawContent</sofaName>
 </inputSofas>
 <outputSofas>
 <sofaName>detagContent</sofaName>
 </outputSofas>
 </capability>
</capabilities>
```

##### ## ##### ##### ##### ## ##### 2,   
 ##   
##### ##### ##### ##### ##### ##### ## ##### 1.9,  
##### ## 

### 6.3. Sofa Capabilities and APIs for Applications

## #####, ##### ## ## ## ## #####. ## ##  
##### # ## ##, ## ##### ## ## ## ## ## ##  
## ## ## ## ## ## #####. ##### ## ## ##  
##### ## ## ##### ## ## ##. ## ## ## ##  
##### ## ## ##### ## ## ##. ## ##### ## ## ##  
##### ## ##### 5, ~~~~~  
~~~~~ [123] ## ##### ## ## ## ##.

```
##, ## ##### ## ## ## ## ##
#_#####. ## ## ## ## ## ##### ## ## ## ## ## (## ##
#####) ## ## ## ## ##### ##.
```

## 6.4. Sofa Name Mapping

```
#####
#####
#####
#####
#####
```

```
#####
#####
#####
#####
```

```
#####
```

```
CAS viewX = cas.createView("X");
```

```
#####
```

```
CAS viewX = cas.getView("X");
```

```
#####
#####
#####
```

```
#####
#####
#####
```

```
#####
#####
#####
```

```
#####
```

```
Iterator allViews = cas.getViewIterator();
```

```
#####
#####
#####
```

```
Iterator someViews = cas.getViewIterator(String namePrefix);
```

```
#####
```

```
#####
6.4.5, ##### [136].
```

## 6.4.1. Name Mapping in an Aggregate Descriptor

```
#####, ##### ##### #####
#####.
```

```
#####. ##### ##### ##### #####
#
#####.
```

```
#####
#####. ##### ##### ##### ##### ##### # #
#####.
```

```
#####.
#####
#####, #####.
```

```
#####, ##### ##### ##### # #####
#####:
```

```
<sofaMappings>
 <sofaMapping>
 <componentKey>SpeechToText</componentKey>
 <componentSofaName>AudioInput</componentSofaName>
 <aggregateSofaName>SegmentedAudio</aggregateSofaName>
 </sofaMapping>
 <sofaMapping>
 <componentKey>SpeechToText</componentKey>
 <componentSofaName>TranscribedText</componentSofaName>
 <aggregateSofaName>EnglishTranscript</aggregateSofaName>
 </sofaMapping>
 <sofaMapping>
 <componentKey>EnglishToGermanTranslator</componentKey>
 <componentSofaName>EnglishDocument</componentSofaName>
 <aggregateSofaName>EnglishTranscript</aggregateSofaName>
 </sofaMapping>
 <sofaMapping>
 <componentKey>EnglishToGermanTranslator</componentKey>
 <componentSofaName>GermanDocument</componentSofaName>
 <aggregateSofaName>GermanTranslation</aggregateSofaName>
 </sofaMapping>
</sofaMappings>
```

```
#####
#####. ##### 1.9.1, ##### (### #####) ##### #####
#####.
```

## 6.4.2. Name Mapping in a CPE Descriptor

```
#####
(##### ### ### #####). #### ##### ## ## #####
#####: <collectionIterator>, <casInitializer>
<casProcessor>. ## ## ##### ##### ## ## #####,
##
(#####, #####, #####). ##### ##
<componentKey> ##### ## ## #####. #####, #####
#####, ## ## ##### ## ##### #####. ## ##### 3.6.1.3,
#<#####> ##### ## ~~~~~.
```

##### ## #####. ##### ## ## ##### ##### ## ## #####  
##### #####. ##### ## ## ## ## ##### ##### ##### #####  
## ##### ##### #####. ##### ## ##### ## ##### #####  
### ##### ##### #####. ##### ## ## ##### ## #####.

```
<collectionReader>
 <collectionIterator>
 <descriptor>
 . . .
 </descriptor>
 <configurationParameterSettings>...</configurationParameterSettings>
 <sofaNameMappings>
 <sofaNameMapping componentSofaName="nextSegment"
 cpeSofaName="SegmentedAudio"/>
 </sofaNameMappings>
 </collectionIterator>
 <casInitializer/>
</collectionReader>
```

```
#####
#####, #####, ##
#####.
```

## 6.4.3. Specifying the CAS View for a Single-View Component

```
#_ #####, ## # ##### ## #####
##.
```

```
#####, ##### ## ## ## ##### ## ## ## ## ##
##, ## ##
##. ##
##
##:
```

```
<casProcessor>
```

```

 . . .
 <sofaNameMappings>
 <sofaNameMapping componentSofaName="_InitialView"
 cpeSofaName="GermanTranslation"/>
 </sofaNameMappings>
 </casProcessor>

```

#####  
#####.

```


#####.
```

```
//create a "root" UIMA context for your whole application

UimaContextAdmin rootContext =
 UIMAFramework.newUimaContext(UIMAFramework.getLogger(),
 UIMAFramework.newDefaultResourceManager(),
 UIMAFramework.newConfigurationManager());

input = new XMLInputSource("test.xml");
desc = UIMAFramework.getXMLParser().parseAnalysisEngineDescription(input);

//setup sofa name mappings using the api

HashMap sofamappings = new HashMap();
sofamappings.put("localName1", "globalName1");
sofamappings.put("localName2", "globalName2");

//create a UIMA Context for the new AE we are about to create

//first argument is unique key among all AEs used in the application
UimaContextAdmin childContext = rootContext.createChild("myAE", sofamap);

//instantiate AE, passing the UIMA Context through the additional
//parameters map

Map additionalParams = new HashMap();
additionalParams.put(Resource.PARAM_UIMA_CONTEXT, childContext);

AnalysisEngine ae =
 UIMAFramework.produceAnalysisEngine(desc,additionalParams);
```

#####  
#####  
#####  
#####.



```
CrossAnnotation #### ## ##### ##### uima.tcas.Annotation ### ##### ##
#####: # ##### ## ##### #####.
```

```
CAS cas = seAnnotator.newCAS();
CAS aView = cas.createView("EnglishDocument");
```

```
aView.setTextDocumentText("this beer is good");
```

```
// get View of the English text Sofa
englishView = aCas.getView("EnglishDocument");

// Create the output German text Sofa
```

```
germanView = aCas.createView("GermanDocument");
```

```
##:
```

```
englishView.addFsToIndexes(engAnnot);
. . .
germanView.addFsToIndexes(germAnnot);
```

```
#####
#####:
```

```
// add link to English text
germAnnot.setFeatureValue(other, engAnnot);
```

## 6.6.4. Accessing the results of analysis

```
#####, ##### ## ## ##
#####. ##### ##### ## ## ## ## ## ##### ##### ## ##### ## ##
##. ## ## ##### ##:
```

```
//get annotation iterator for this CAS
FSIndex anIndex = aView.getAnnotationIndex();
FSIterator anIter = anIndex.iterator();
while (anIter.isValid()) {
 AnnotationFS annot = (AnnotationFS) anIter.get();
 System.out.println(" " + annot.getType().getName()
 + ": " + annot.getCoveredText());
 anIter.moveToNext();
}
```

```
##, ##### ## ## ## #####:
```

```
if (annot.getType() == cross) {
 AnnotationFS crossAnnot =
 (AnnotationFS) annot.getFeatureValue(other);
 System.out.println(" other annotation feature: "
 + crossAnnot.getCoveredText());
}
```

```
getCoveredText().
#####
#####. ## ## ## ##### ## ## ## #####
#####.
```

```
##:
```

```
---Printing all annotations for English Sofa---
uima.tcas.DocumentAnnotation: this beer is good
uima.tcas.Annotation: this
```



```

uima.tcas.Annotation: beer
uima.tcas.Annotation: is
uima.tcas.Annotation: good

---Printing all annotations for German Sofa---
uima.tcas.DocumentAnnotation: das bier ist gut
sofa.test.CrossAnnotation: das
 other annotation feature: this
sofa.test.CrossAnnotation: bier
 other annotation feature: beer
sofa.test.CrossAnnotation: ist
 other annotation feature: is
sofa.test.CrossAnnotation: gut
 other annotation feature: good

```

## 6.7. Views API Summary

```

Single-View ##### ##
##/## #####

```

```

Multi-View ##### ## #####, ## ##### ##### ## ## ## ## ##
#:

```

```

#:

```

```

JCas newView = aJCas.createView(String localNameOfTheViewBeforeMapping);
CAS newView = aCAS.createView(String localNameOfTheViewBeforeMapping);

```

```

#:

```

```

JCas myView = aJCas.getView(String localNameOfTheViewBeforeMapping);
CAS myView = aCAS.getView(String localNameOfTheViewBeforeMapping);
Iterator allViews = aCasOrJCas.getViewIterator();
Iterator someViews = aCasOrJCas.getViewIterator(String localViewNamePrefix);

```

```

#:

```

```

#:

```

```

aCasOrJCas.setDocumentText(String docText);
aCasOrJCas.setSofaDataString(String docText, String mimeType);
aCasOrJCas.setSofaDataArray(FeatureStructure array, String mimeType);
aCasOrJCas.setSofaDataURI(String uri, String mimeType);

```

```

#:

```

```

String doc = aCasOrJCas.getDocumentText();
String doc = aCasOrJCas.getSofaDataString();
FeatureStructure array = aCasOrJCas.getSofaDataArray();
String uri = aCasOrJCas.getSofaDataURI();

```

```
InputStream is = aCasOrJCas.getSofaDataStream();
```

## 6.8. Sofa Incompatibilities between UIMA version 1 and version 2

```
2 ## ##### ## ## ##### ## ##### ## #####
#####. ##### ## ##### ##, ae, ## ##
```

```
CAS cas = ae.newCas();
```

```
##. ## ## ##### # #### ## ## ## ## ## # _ #####.
##. ##
##
#####.
```

```
##, ##### ## ##
##. ##### ## ## ## ##
##.
```

```
##, ## ##### ##### ## ## ## ## #2.0 #####:
```

- ## ## ## 1.# ##### (## ## ## ## ## ## ## 2).
- ##### ## ##### ##### ##### # \_ ##### ##  
#### ## #####.
- ##### ##### ## ## ## ## ## ## ##.

## Chapter 7. CAS Multiplier Developer's Guide

### ##### (#####) #####  
## ##### ## ## # #####, ##### ## ##,  
### #####. ##### ## #####  
##### ## ## ## #2.0: # ## ## ##### #  
#####  
#####  
##### ## ##### # #####. ##  
## # ##### ## ##### ## #####, ## ## ##  
## ##### ## #####. # ## ##### ## ##  
### ##### (##### ## # ##### ## ## ##  
### ##### 5.2, ##### ## ##### [123]) ## ##### # ##### ## ##  
##### ## ##### ## ## #####.

### ##### ## ## ##### ## ##### ## ##, #####.  
# ## ##### ## ## ## ## ##### ##### ## ##  
#####. ## #####, # ## ##### ## ##  
##### ## #####; ##### #, ## ##### ## ## ## ##  
#####.

## 7.1. Developing the CAS Multiplier Code

### 7.1.1. CAS Multiplier Interface Overview

```
JCasMultiplier_ImplBase
CasMultiplier_ImplBase #####, ##### ## ##### ##### ##### ## ###.
#####, ### ## ##### #####
initialize, destroy, ### reconfigure #####. ##### ## #####
#####: process, hasNext, ### next. ### ##### ##### ##### ##
#####:
```

1. ### #####  
###. ### #####, ### ### ## # ##### ## ### ##### ##.
2. ### ##### hasNext #####. ### ##  
##### true ##### ## ## ##### ## ##### ## ##  
#### (### #####, ##### ## #### ##), ### false ## ##.
3. ## hasNext #####, ### ##### next  
#####. ### ## ##### # ## ## (## #### ## ## # ##),  
##### ##, ### ##### ## #### ## next #####.
4. ##### 2 ## 3 ##### hasNext #####.

```
process ## ##### ## ## hasNext ##### ##### ## ##, ##
process #####. ## ##
```

```
#####: ## ##### ## #### ## ##### ##### ##### ## #### ##
#####. ## ##### ##### ##### ##### # ##### ##### ## #### ##
##. ## #### ##### ## # #####
```

```
##. ## ### ##### ## ## ##,
#####.
```

## 7.1.3. Example Code

```
#####
#####.### ##### ## ## #####
org.apache.uima.examples.casMultiplier.SimpleTextSegmenter ### ## #####
examples/src #####.
```

### 7.1.3.1. Overall Structure

```
public class SimpleTextSegmenter extends JCasMultiplier_ImplBase {
 private String mDoc;
 private int mPos;
 private int mSegmentSize;
 private String mDocUri;

 public void initialize(UimaContext aContext)
 throws ResourceInitializationException
 { ... }

 public void process(JCas aJCas) throws AnalysisEngineProcessException
 { ... }

 public boolean hasNext() throws AnalysisEngineProcessException
 { ... }

 public AbstractCas next() throws AnalysisEngineProcessException
 { ... }
}
```

```
SimpleTextSegmenter ##### JCasMultiplier_ImplBase ###
initialize ##### ## ## ## ## ##### process, hasNext, ## next
#####. ##### ## #####.
```

### 7.1.3.2. Initialize Method

```
public void initialize(UimaContext aContext) throws
 ResourceInitializationException {
 super.initialize(aContext);
 mSegmentSize = ((Integer)aContext.getConfigParameterValue(
 "segmentSize")).intValue();
}
```

```
#####, # ## ##### ## ##### ## ##### ## ##
#####.###
#####, ##### ## ## ##### ## ##### ## ## (## #####)
##.
```



```

 breakAt = mDoc.length();

 // search for the next newline character.
 // Note: this example segmenter implementation
 // assumes that the document contains many newlines.
 // In the worst case, if this segmenter
 // is run on a document with no newlines,
 // it will produce only one segment containing the
 // entire document text.
 // A better implementation might specify a maximum segment size as
 // well as a minimum.

 while (breakAt < mDoc.length() &&
 mDoc.charAt(breakAt - 1) != '\n')
 breakAt++;

 JCas jcas = getEmptyJCas();
 try {
 jcas.setDocumentText(mDoc.substring(mPos, breakAt));
 // if original CAS had SourceDocumentInformation,
 // also add SourceDocumentInformation
 // to each segment
 if (mDocUri != null) {
 SourceDocumentInformation sdi =
 new SourceDocumentInformation(jcas);
 sdi.setUri(mDocUri);
 sdi.setOffsetInSource(mPos);
 sdi.setDocumentSize(breakAt - mPos);
 sdi.addToIndexes();

 if (breakAt == mDoc.length()) {
 sdi.setLastSegment(true);
 }
 }

 mPos = breakAt;
 return jcas;
 } catch (Exception e) {
 jcas.release();
 throw new AnalysisEngineProcessException(e);
 }
 }
}

```

```

next
next ##### hasNext
process ## next .

```

```

#####
#####. #####:

```

```

JCas jcas = getEmptyJCas();

```

```
#####, ##### ##### # #### ##
#####.
```

```
####, ##### ## ## ## try...catch ##### ## ##### ##### # #### ## ##### ##### ##
#####. ##### ## ##### ##### ## ##### # ### #####
#####.
```

## 7.2. Creating the CAS Multiplier Descriptor

```
#####. ##### #####
#####, ## ## ##### ##### ##### ## #####
#####.
```

```
SimpleTextSegmenter ## ##### ## ## examples/
descriptors/cas_multiplier/SimpleTextSegmenter.xml ##### ## ## ##.
```

```
#####, ## ## ##### #####, ##
#####. ## ## #####
#####, ##### ##### ## ## ##.
```

```
##, ## ## ## ##### ##### ##### ## ##
#####, ## ## ##:
```

SimpleTextSegmenter.xml

### Overview

▼ **Implementation Details**

Implementation Language ☐ C/C++ ☒ Java

Engine Type ☒ Primitive ☐ Aggregate

▼ **Runtime Information**

This section describes information about how to run this component

☐ updates the CAS

☒ multiple deployment allowed

☒ Outputs new CASes

Name of the Java class file

```
##, ## ## ##
<outputsNewCASes> ##### ## #####:
```

```
<operationalProperties>
 <modifiesCas>false</modifiesCas>
 <multipleDeploymentAllowed>true</multipleDeploymentAllowed>
 <outputsNewCASes>true</outputsNewCASes>
</operationalProperties>
```



### 7.3. Using a CAS Multiplier in an Aggregate Analysis Engine

### 7.3.1. Adding the CAS Multiplier to the Aggregate

### 7.3.2. CAS Multipliers and Flow Control

### ##### ## ## ##### ## ##### ## ##### ## #####  
## ## ## ## ##### ##### ## ## ##### ##### ## ##  
##### ## ## ##### ##### ## ## ##. ##### ## ## ##:  
#### # ## # ## ## ##### #, #### ## # ##### # ##  
##### ## ## ##### ##, ##### ## ## ##### # ##  
##### ## ## ##### ##, ##### ## ## ##### # ## #####.

#### ##### # ## #####, ## ##### ##### ##### ## ## ##, ####  
#### ##### ## ## ## ## ##### ## ## ##### ##### ## ##, ##### #  
## ## ##### ##### ## ## ##### # ## #####. # #####  
#### # ## ## ##### ##### ## ##### # ## ##### # ## ##  
✂✂✂ ##### # ## ##.

## ## ## ##### ✂✂✂ ##### ## ##### ## # ## ## ##, ####  
#### ##### ✂✂✂✂ ##### ## ## ##. ##### ## #####, ## ##  
#### # ## ##### ## ## ##### # ##### ## ## ## ## ## ## ## ##  
##### ## # ##### # ##### ##.

```
<configurationParameters>
 <configurationParameter>
 <name>ActionForIntermediateSegments</name>
 <type>String</type>
 <multiValued>false</multiValued>
 <mandatory>false</mandatory>
 <overrides>
 <parameter>
 FixedFlowController/ActionAfterCasMultiplier
 </parameter>
 </overrides>
 </configurationParameter>
</configurationParameters>

<configurationParameterSettings>
 <nameValuePair>
 <name>ActionForIntermediateSegments</name>
 <value>
```

```
<string>drop</string>
</value>
</nameValuePair>
</configurationParameterSettings>
```

```


examples/descriptors/cas_multiplier/Segment_Annotate_Merge_AE.xml.### ###

#####,### ##### 4.3, ##### ##### [118].

4,
[113]. ### #####

#####
```

```


##,#####
####<outputsNewCASes>#####
#####.#####
#####7.2,##### [146].
```

```
true,### ## ## ## ## ## ## ## ## ##

#####
```

```
<#####> ##### ## false ,### ## ## #####
(#.#.### #####
####)#### #### #### ##### #####.#### ##

#####;### #### ## ##### ## ##### ## ## ##### ## ## ##
#####.
```

#####: ## ### ##### ## ##### ##### ## ##### ##### ## ## #####, ## ##### ##  
##### # ##### ##### ##### ##### ##### ##### # ## ##### 4.5,  
##### ##### ##### ## ##### [119].

```
#
#####. ##### # # # # # # # # # # # # # #
```

```
outputsNewCASes #####
false, ##### ## ##### ##### ##### ##### ## ## ##### ##### ## ##.

#####
#####, ##### ## ## #####. ##### ##### ##### ## ## ##### ##, ##
##.
```

## 7.5. Calling a CAS Multiplier from an Application

### 7.5.1. Retrieving Output CASes from the CAS Multiplier

```
AnalysisEngine ##### ## ## ##### ##### ##### ## ##
#####:

 • CasIterator processAndOutputNewCASes(CAS)

 • JCasIterator processAndOutputNewCASes(JCas)

#####, ## ## processAndOutputNewCASes ## ## ## ## ##.
##
#####.
```

```
#####
(## ##### ## CAS.release() #####) ##### ## #####
~~~~~ ## ##### ## CasIterator.next ##### ##. #####,
##.
```

```
org.apache.uima.examples.casMultiplier.
CasMultiplierExampleApplication ##### ## ## ## ##### ##:
```

```
CasIterator casIterator = ae.processAndOutputNewCASes(initialCas);
while (casIterator.hasNext()) {
 CAS outCas = casIterator.next();

 //dump the document text and annotations for this segment
 System.out.println("***** NEW SEGMENT *****");
 System.out.println(outCas.getDocumentText());
 PrintAnnotations.printAnnotations(outCas, System.out);

 //release the CAS (important)
 outCas.release();
}
```

```
7.1.1,
[141], ## ## ##### ## ## ## ## (initialCas
#####) ##### ## ## ## ## ## ##. #####
initialCas #####
CasIterator.hasNext ##### ## ## ## ##, ##### ## ## ## ##
#####.
```

```
#####
CasIterator's hasNext ## next #####. ##### ##, ## ##### #####
##. ##
CasIterator ##
#####.
```

```
####, ##### ## ## CasIterator ## ##### ##### ##### ## ##### ## #####
#####. ## ## ##### ## #####, ## ##### ## ##
####, ## ## ##### ##### ##### ## #####. ##### ## ##### ## #####
#####.
```

## 7.5.2. Using a CAS Multiplier with other Analysis Engines

```
#####
process ##### ## ##### #####. ##### #####
#####.
```

```
#####, ## ##### ##### ## # ## ##### ##### ##### # #####
#####
#####.
#####,
#####, ##### ## ##### ##### ## ##
#####
#####.
```

```
#
UimaContext ##### ## #####, ## #####:
```

```
//create a "root" UIMA context for your whole application

UimaContextAdmin rootContext =
 UIMAFramework.newUimaContext(UIMAFramework.getLogger(),
 UIMAFramework.newDefaultResourceManager(),
 UIMAFramework.newConfigurationManager());

XMLInputSource input = new XMLInputSource("MyCasMultiplier.xml");
AnalysisEngineDescription desc = UIMAFramework.getXMLParser().
 parseAnalysisEngineDescription(input);

//create a UIMA Context for the new AE we are about to create

//first argument is unique key among all AEs used in the application
UimaContextAdmin childContext = rootContext.createChild(
 "myCasMultiplier", Collections.EMPTY_MAP);

//instantiate CAS Multiplier AE, passing the UIMA Context through the
//additional parameters map

Map additionalParams = new HashMap();
```

```
additionalParams.put(Resource.PARAM_UIMA_CONTEXT, childContext);

AnalysisEngine casMultiplierAE = UIMAFramework.produceAnalysisEngine(
 desc, additionalParams);

//repeat for another AE
XMLInputSource input2 = new XMLInputSource("MyAE.xml");
AnalysisEngineDescription desc2 = UIMAFramework.getXMLParser().
 parseAnalysisEngineDescription(input2);

UimaContextAdmin childContext2 = rootContext.createChild(
 "myAE", Collections.EMPTY_MAP);

Map additionalParams2 = new HashMap();
additionalParams2.put(Resource.PARAM_UIMA_CONTEXT, childContext2);

AnalysisEngine myAE = UIMAFramework.produceAnalysisEngine(
 desc2, additionalParams2);
```

#####  
#####.

- ```

1. ##### process #####,###
##### (#####)
#####.###
org.apache.uima.util.CasCopier #####
#####.

2. ##### hasNext #####,###
##### false ## ## ## ##.

3. ##### process ##### # ## ##,###
##### ## ## ##, #####
#####.

4. #####,### ## ## ## ## ##
##,## true ## hasNext ## ##
##### next ##,### ## ## ## ##.

```


#####

7.6.2. Example CAS Merger

```
## ##### ### ##### ##### ##### ##### ## ##### ##
##### ## ### ##### ##. ### ##### ##### ##### ##### ##
org.apache.uima.examples.casMultiplier.SimpleTextMerger ### ## ##### #####
## ##### ##### ## examples/src #####.
```

7.6.2.1. Process Method

```
##### ## # ## # ## # ## # ##### # # # process #####, ### ##### # # #  
process ##### # ## # ## # ##### # ## # # ## # # # # ## #  
####:
```

```

public void process(JCas aJCas) throws AnalysisEngineProcessException {
    // procure a new CAS if we don't have one already
    if (mMergedCas == null) {
        mMergedCas = getEmptyJCas();
    }

    // append document text
    String docText = aJCas.getDocumentText();
    int prevDocLen = mDocBuf.length();
    mDocBuf.append(docText);

    // copy specified annotation types
    // CasCopier takes two args: the CAS to copy from.
    //                               the CAS to copy into.
    CasCopier copier = new CasCopier(aJCas.getCas(), mMergedCas.getCas());

    // needed in case one annotation is in two indexes (could
    // happen if specified annotation types overlap)
    Set copiedIndexedFs = new HashSet();
    for (int i = 0; i < mAnnotationTypesToCopy.length; i++) {
        Type type = mMergedCas.getTypeSystem()
            .getType(mAnnotationTypesToCopy[i]);
        FSIndex index = aJCas.getCas().getAnnotationIndex(type);
        Iterator iter = index.iterator();
        while (iter.hasNext()) {
            FeatureStructure fs = (FeatureStructure) iter.next();
            if (!copiedIndexedFs.contains(fs)) {
                Annotation copyOfFs = (Annotation) copier.copyFs(fs);
                // update begin and end
                copyOfFs.setBegin(copyOfFs.getBegin() + prevDocLen);
                copyOfFs.setEnd(copyOfFs.getEnd() + prevDocLen);
                mMergedCas.addFsToIndexes(copyOfFs);
                copiedIndexedFs.add(fs);
            }
        }
    }
}

```

```

### CasCopier ##### ## ##### ## ##### ##### ##### ##### ##### (#####
## # ##### #####) ## ## ##### ##. ### CasCopier ##### #####,
##### ## ## ## ##### ##### ##### ##### ##### #####, ###
##### ##### ##### ## #####.

##### ##### ##### ## ##### ##### # ##### StringBuffer. #####
## ##### ##### ##### ## ## ##### ## ## ##### ## #####
##### ## ##### ## ## ##.

### ##### ## ## process ##### ##### ## ## ## ## ## ##.
### #####, ## ## ##### ## ##### ## ##### ## ## #####
#####. ##### ## ## ## ##### ## SourceDocumentInformation #####
## ## ## ## ## ## lastSegment ##### ## ## ## true. ##### (##### ## ##
## ## ##### SimpleTextSegmenter #####) ##### ## ## ## ##
##### ## ## #####, ## ##### ## ##### ##### ##### ## ## ##
##### ## ##### ## ##### ##.

// get the SourceDocumentInformation FS,
// which indicates the sourceURI of the document
// and whether the incoming CAS is the last segment
FSIterator it = aJCas
    .getAnnotationIndex(SourceDocumentInformation.type).iterator();
if (!it.hasNext()) {
    throw new RuntimeException("Missing SourceDocumentInformation");
}
SourceDocumentInformation sourceDocInfo =
    (SourceDocumentInformation) it.next();
if (sourceDocInfo.getLastSegment()) {
    // time to produce an output CAS
    // set the document text
    mMergedCas.setDocumentText(mDocBuf.toString());

    // add source document info to destination CAS
    SourceDocumentInformation destSDI =
        new SourceDocumentInformation(mMergedCas);
    destSDI.setUri(sourceDocInfo.getUri());
    destSDI.setOffsetInSource(0);
    destSDI.setLastSegment(true);
    destSDI.addToIndexes();

    mDocBuf = new StringBuffer();
    mReadyToOutput = true;
}

##### ## ## ##### ## ##### ##, ### ##### ##### #####
### ##### ## (##### ## ##### ## ## ##### # SourceDocumentInformation
#####), ### ##### ## mReadyToOutput ##### ## #####. ##### ##
##### ## ## hasNext ## next #####.

```



```
## ##### ##### ## ##### ##### ##### ##### ##
SimpleTextMerger ## ##### ## examples/descriptors/cas_multiplier/
Segment_Annotate_Merge_AE.xml. #### ##### ##### ##
SimpleTextSegmenter ##### ## ##### # ##### ##### ##### #####. ## #####
##### ##### ## ##### ##### ##### ##### #####. #####
## ##### ## SimpleTextMerger ## ##### ## ##### ## ## ##. ## Name
##### ## ##### ## ## ##### ##### ## ## ## Token ##### ## ##.

##### ##### ## ## ## ##### ##### ##### ##### ## ##
##### ##### ## ## ##### # ##### ##### ## ##### ## ##
##### ## ##### ## ## #####. ##### ##### ## ##, ##### ##
##### # ## ## ##, ##### ## ## ##### ## ## ##### ##### #####.
```


8.3. Using XMI CAS Serialization

158

```
##### ## ### ##### ## ##### ##### ## ##### ##
#####, ##### ## ## ## ##### #####.

org.apache.uima.examples.xml.XmiCollectionReader: #### # #
##### ##### ##### # ##### ## ## ##### #####
#### # #### # ##. ####, #### ##### ## # #### #
##### ##### ##### ##### ##### #, ##### #####
#### ##### #####, ## ## # ##### #####.

#####, ## ##### ## ##### uimaj-examples/ecore_src ## ## #####
org.apache.uima.examples.xml.XmiEcoreCasConsumer, ##### ##### ##
## ## ##### ## ##### ## ## ##### ## ## #####. ##### ##### ##
UimaTypeSystem2Ecore #####, ## ##### # ## ##### # ## #####
## ## #####.###, #####.###, ## #####.###.### # ## #####_###/#####.### ##
#####.
```

8.3.1. Character Encoding Issues with XML Serialization

```
#### ##### ## ## ##### ##### ##### ## ##### ##, ## ##### ## #
### 1.0. #####, ## ## ##### ## ##### ##### ## ##### ## ## #####
##### #####, ## ##### ## #####. ## ##### ##### ## #####
##### ##### ## ##### ##### ## #####, #### ## ## ##. ####
##### ##### ## ## ##### ##### ## ## ## ##### ## ##### ## ##
##### ## ## ##### ##. ## ##### ##### ##### ## ##### #####
#####, ## ##### ## ##### ## ##### ##### ##### (# #####
##### ## ##). #### ## ## ##### ##### ## ##### #####
##### #####; ## ##### ##### ## ## #####
#### ## ## ## ##### #####.
```

```
## ## ##### ## ## ##### ## ## ## #####, ## ## ##### ## ## #####
##### ## #####, ## ##### ##### ## ##### ##### ## ## #####
## #####. ##### ##### ##### ##### ## #####, ## ##### ## #####
## #####, ## ## ## ## #####. ## ## # ##### ##### ##
##### ## ##### ##### ## ## ##### ##### ## #####.
```

```
#### ##### # ##### ## ##### ## ##### ## #
##### ##### #####. ##### ##### ## ##### ##
org.apache.uima.internal.util.XMLUtils.checkForNonXmlCharacters(),###
#####. ##### ##### ## ##### ## #####.
```

```
##### ## ## ##### ## ## ##### ## ## #####, #### ## ##
##### ## ##### ## ## ## ##.
```

