

# Creating new Resource Types

## Additional topics

Third GATE Training Course  
August–September 2010

© 2010 The University of Sheffield

This material is licenced under the Creative Commons

Attribution-NonCommercial-ShareAlike Licence

(<http://creativecommons.org/licenses/by-nc-sa/3.0/>)

## Corpus-level processing

- When running a PR over a corpus of more than one document, you may want to do some additional pre- and post-processing before the first and after the last document.
- To do this, implement `gate.creole.ControllerAwarePR`
- Three callback methods called at key points in the execution of the *controller that contains the PR*:
  - `controllerExecutionStarted`
  - `controllerExecutionFinished`
  - `controllerExecutionAborted`
- Parameter is the `Controller`.
- “aborted” callback also receives the `Throwable` that caused the abort.

- So if the controller is a `CorpusController`, these correspond to:
  - before the first document
  - after the last document
  - when something goes wrong

## ControllerAwarePR example

```
1 public class ExampleAnalyser
2     extends AbstractLanguageAnalyser
3     implements ControllerAwarePR {
4     public void controllerExecutionStarted(Controller c) {
5         if(c instanceof CorpusController) {
6             System.out.println("Processing corpus " +
7                 ((CorpusController)c).getCorpus().getName());
8         }
9         else {
10            System.out.println(
11                "Running in a simple pipeline");
12        }
13    }
14
15    // controllerExecutionFinished is similar
16 }
```

## Exercise 1: Corpus statistics

Add corpus statistics to your DocStats PR:

- Add private fields to keep a running total count of words (and nouns/verbs).
- Implement `ControllerAwarePR`.
- In the “started” callback, initialize these totals to 0.
- In the “finished” callback
  - check whether you are running in `CorpusController`
  - if so, put the total counts into features on the controller’s `Corpus`.
- You can leave the “aborted” callback empty (or just print a message).

Example solution on the wiki.

# Adding actions to the GUI

- Any (language, processing or visual) resource can contribute *actions* to the GATE developer GUI.
- These appear as items on the resource's right-click menu. For example:
  - The “Run” option for controllers comes from the controller editor VR
  - The “Save as...” and “Delete ontology data” options for an ontology LR come from the LR itself.
- This is done by implementing the interface  
`gate.gui.ActionsPublisher`
- One method, returning a `List` of `javax.swing.Action` objects.

## Exercise 2: ActionsPublisher

Implement cumulative statistics for your DocStats PR:

- keep a running total as before, but rather than resetting it in `controllerExecutionStarted`, provide an action to reset it explicitly.
- provide another action to display the current total.
- example code available on the wiki.