Crawling Assignment

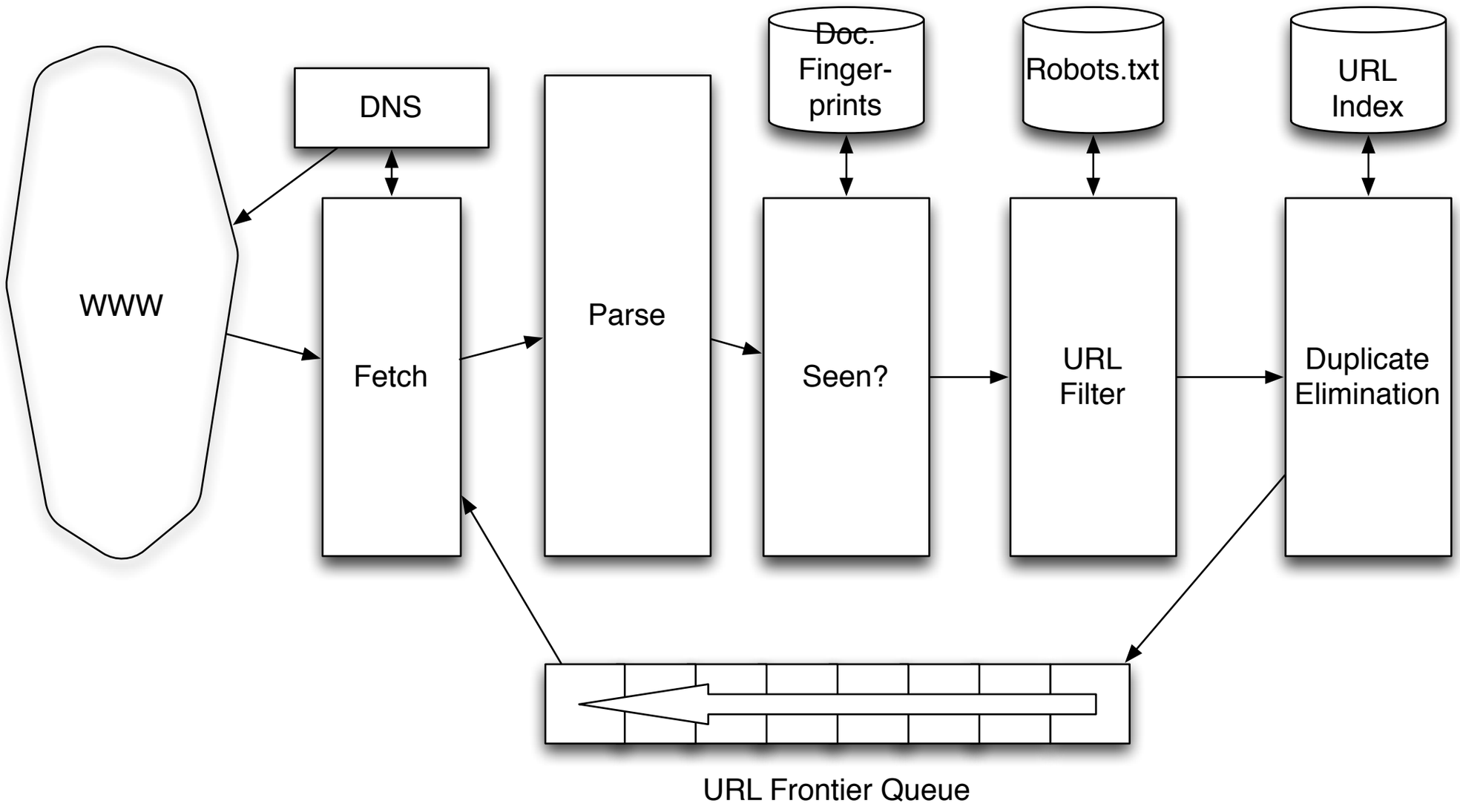
Introduction to Information Retrieval CS 150

Donald J. Patterson

Content adapted from Hinrich Schütze
http://www.informationretrieval.org

Robust Crawling

A Robust Crawl Architecture



Crawling Assignment

Setting up Eclipse

- Create a new java project in Eclipse

Crawling Assignment

Setting up Eclipse

- Create a new java project in Eclipse
- In your src folder, put your code from the Text Processing assignment

Crawling Assignment

Setting up Eclipse

- Create a new java project in Eclipse
- In your src folder, put your code from the Text Processing assignment
- Download crawler4j’s code (yasserg on github)
  - Maven
  - crawler4j and crawler4j's dependencies
  - crawler4j with dependences

Crawling Assignment

Setting up Eclipse

- Create a "lib" folder as a sibling to your "src" folder

Crawling Assignment

Setting up Eclipse

- Create a "lib" folder as a sibling to your "src" folder
- Put all the jars into the "lib" folder

Crawling Assignment

Setting up Eclipse

- Create a "lib" folder as a sibling to your "src" folder
- Put all the jars into the "lib" folder
- Refresh Eclipse ("F5"), so that Eclipse becomes aware of the new files on your hard disk

Crawling Assignment

Setting up Eclipse

- Right-click on your project name and select "Properties"
- Choose "Java Build Path"
- Select the "Libraries" Tab
- Select "Add jars" and add all the jars that you downloaded to your project. This allows your code to use the methods and objects that are defined in those jar files

Cr

Se

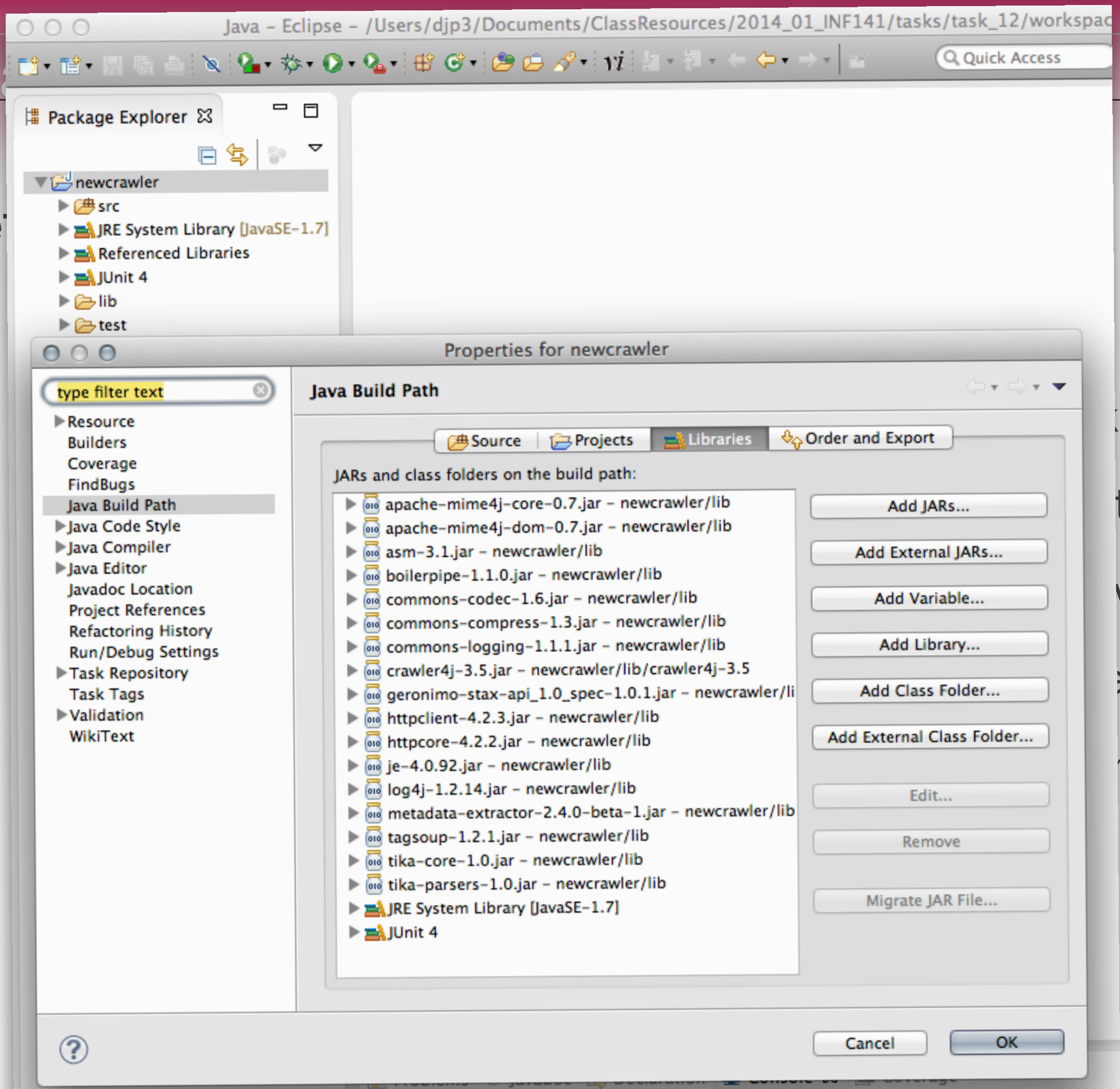
- In

ve

en

t tell

hit



Crawling Assignment

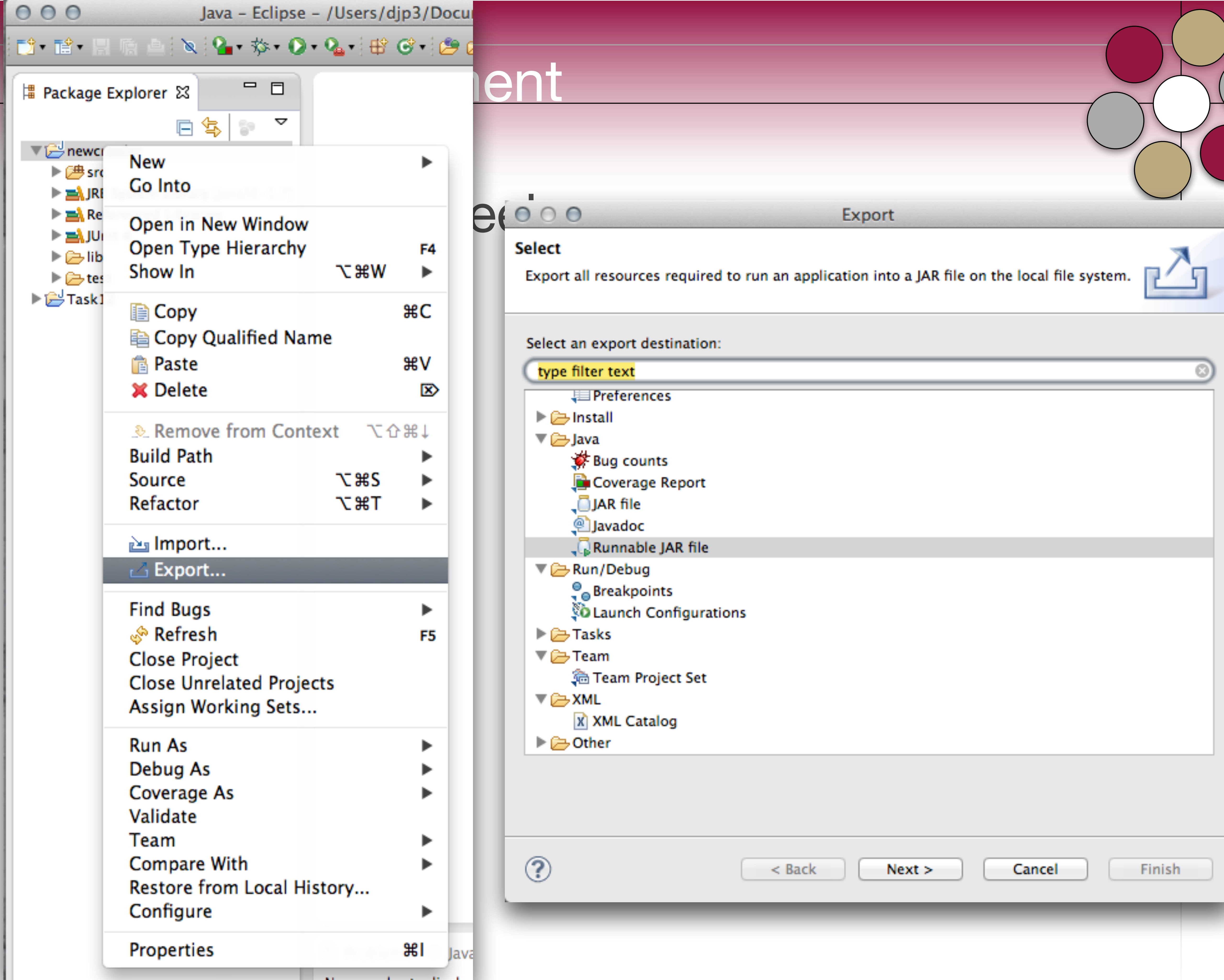
Running on wcpkneel

- First you must have an account on wcpkneel
  - Prof. Patterson has to set that up for you

Crawling Assignment

Running on wcpkneel

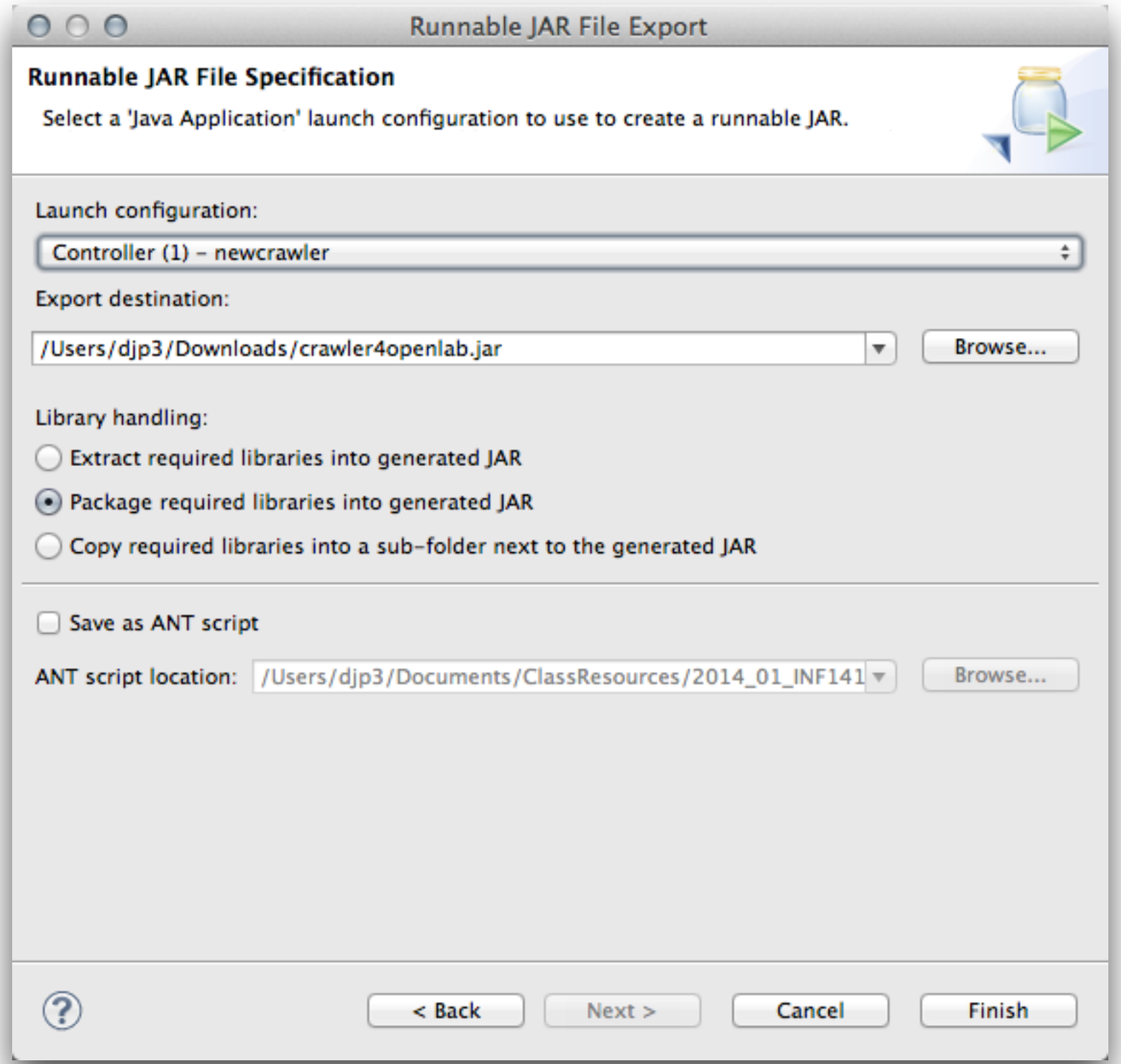
- Export as a runnable jar
  - Run your program on your local machine to create a run profile
  - Export your project as a "Runnable Jar”
    - A runnable jar packages all the required libraries into one big .jar file so you don't have to deal with moving them separately. That's convenient, but it probably violates licensing agreements. Don't do that with code you plan on distributing.
- The launch configuration is asking what "main" you want the runnable jar to start from. If you tested your program in Eclipse, you should have an option in the dropdown



Java – Eclipse – /Users/djp3/Docu
Package Explorer
newc
src
JRE
Re
JU
lib
tes
Task1
New
Go Into
Open in New Window
Open Type Hierarchy F4
Show In ⌥⌘W
Copy ⌘C
Copy Qualified Name
Paste ⌘V
Delete
Remove from Context ⌥⇧⌘↓
Build Path
Source ⌥⌘S
Refactor ⌥⌘T
Import...
Export...
Find Bugs
Refresh F5
Close Project
Close Unrelated Projects
Assign Working Sets...
Run As
Debug As
Coverage As
Validate
Team
Compare With
Restore from Local History...
Configure
Properties ⌘I
Export
Select
Export all resources required to run an application into a JAR file on the local file system.
Select an export destination:
type filter text
Preferences
Install
Java
Bug counts
Coverage Report
JAR file
Javadoc
Runnable JAR file
Run/Debug
Breakpoints
Launch Configurations
Tasks
Team
Team Project Set
XML
XML Catalog
Other
< Back
Next >
Cancel
Finish

Crawling Assignment

Running on wcpkneel



Crawling Assignment

Running on wcpkneel

- Make sure you update any hardcoded file path names to reflect the arrangement on wcpkneel

Crawling Assignment

Running on wcpkneel

- Now you need to open a terminal window on wcpkneel
- I use "ssh" to login to the remote machine. If you are on Windows you will need to use PuTTY

Crawling Assignment

Running on wcpkneel

- Now you have to move the jar you just made to wcpkneel
- On a Mac you can do that with "sftp" from a Terminal. On windows use WinSCP
- Now you need to open a terminal window on wcpkneel
- I use "ssh" to login to the remote machine. If you are on Windows you will need to use PuTTY

Crawling Assignment

Running on wcpkneel

- Once there use "ls" to see if your jar made it okay
- Then make sure you are using the right version of java.
  - java -version

Crawling Assignment

Running on wcpkneel

- Then use the command
  - nice -n 19 java -Xmx1024M -jar <myJar.jar> <args>
  - to start your crawler.
- nice is a command that reduces resource usage
- The -Xmx command tells java how much memory to use while crawling. You shouldn't need too much because the frontier is kept on disk.

Crawling Assignment

Running on wcpkneel

- Once you are happy everything is going okay, detach from the screen with Ctrl-A then "d"
- This will send you back to the real terminal
- If you logout with the "exit" command then your crawler is still running away. You need to make sure you are monitoring it.

Crawling Assignment

Running on wcpkneel

- To go back to it, log in to wcpkneel
- Type "screen -r" and you will reattach to the virtual screen that your crawler is running in. If you hit Ctrl-C it will kill your process, or you can just look at any output that you are generating. Hit Ctrl-A then "d" to detach again, as often as you like

Using Crawler4j

What is it?

- crawler4j is an open source web crawler for Java which provides a simple interface for crawling the Web. Using it, you can setup a multi-threaded web crawler in few minutes.

Using Crawler4j

How do you use it

- You need to create a crawler class that extends WebCrawler.

  This class decides which URLs should be crawled and

  handles the downloaded page.

```
public class TestCrawler extends WebCrawler
```

Using Crawler4j

WebCrawler Class

- This method receives two parameters.
  - The page in which we have discovered this new url
  - The second parameter is the new url.
- You should implement this function to specify whether the given url should be crawled or not
- Based on the destination URL
- Based on the file type
- Based on the referring page
- Based on whatever logic you want to use.

```
@Override
public boolean shouldVisit(Page referringPage, WebURL url) {
    return true;
}
```

Using Crawler4j

WebCrawler Class

- This function is called when a page is fetched and ready to be processed by your program.

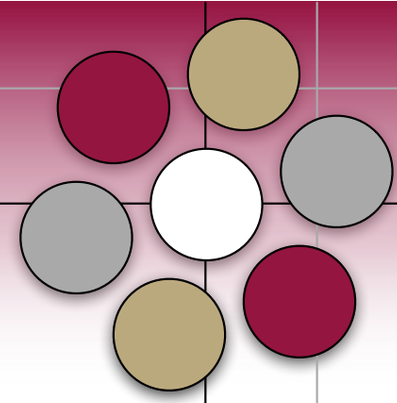
```
@Override
public void visit(Page page) {

}
```

Using Crawler4j

WebCrawler Class

- Useful methods in Page
  - `getWebURL().getURL();`
  - `getParseData();`
    - returns a type based on the data in the page
    - `HtmlParseData`
      - `getText();`
      - `getHtml();`
      - `getOutgoingUrls();`

Using Crawler4j



Controller Class

- Contains the main that you want to run

```
CrawlConfig config = new CrawlConfig();
```

Using Crawler4j

Controller Class

- Specify where to keep the queues, the frontier, the robots files, etc.

```
config.setCrawlStorageFolder(crawlStorageFolder);
```

- “/share/bigspace/CS150-1-F15/<username>”

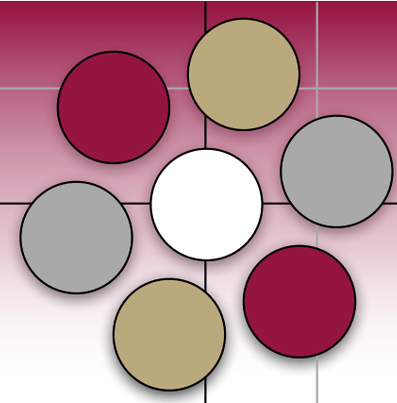
Using Crawler4j

Controller Class

- Set up the components of the crawl

```
PageFetcher pageFetcher = new PageFetcher(config);
RobotstxtConfig robotstxtConfig = new RobotstxtConfig();
RobotstxtServer robotstxtServer = new RobotstxtServer(robotstxtConfig, pageFetcher);
CrawlController controller = new CrawlController(config, pageFetcher, robotstxtServer);
```

Using Crawler4j

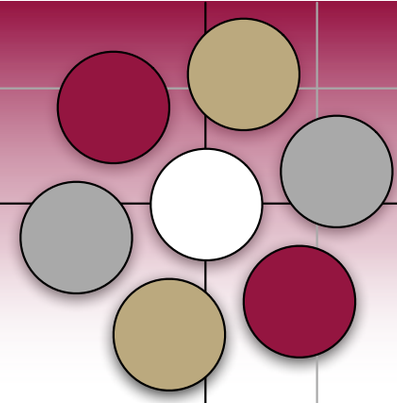


Controller Class

- Add a seed URL
  - Repeat to add more

```
controller.addSeed("http://www.westmont.edu/");
```

Using Crawler4j



Controller Class

- Start the crawl
  - This is a blocking operation

```
controller.start(MyCrawler.class, numberOfCrawlers);
```

Using Crawler4j

Configuration Options

- Set a maximum depth of crawling
  - 0 crawls only the seed pages

```
config.setMaxDepthOfCrawling(maxDepthOfCrawling);
```

Using Crawler4j

Configuration Options

- Set a maximum depth of crawling

```
config.setMaxPagesToFetch(maxPagesToFetch);
```

Using Crawler4j

Configuration Options

- If you suspect that you might crash or have a long running crawl, you can inform crawler4j to maintain state and restart

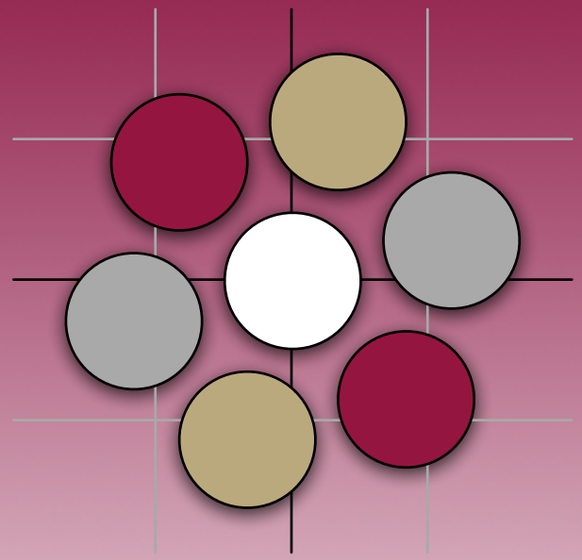
```
config.setResumableCrawling(true);
```

Using Crawler4j

Configuration Options

- User-agent string

```
config.setUserAgentString(userAgentString);
```



WESTMONT COMPUTER SCIENCE