Tutorial 1: Bogor

CISC422/853

Scott Grant

Overview

- Installing Bogor
- Starting Bogor
- General Usage and Tips
- Advice for Assignment 1

Installing Bogor

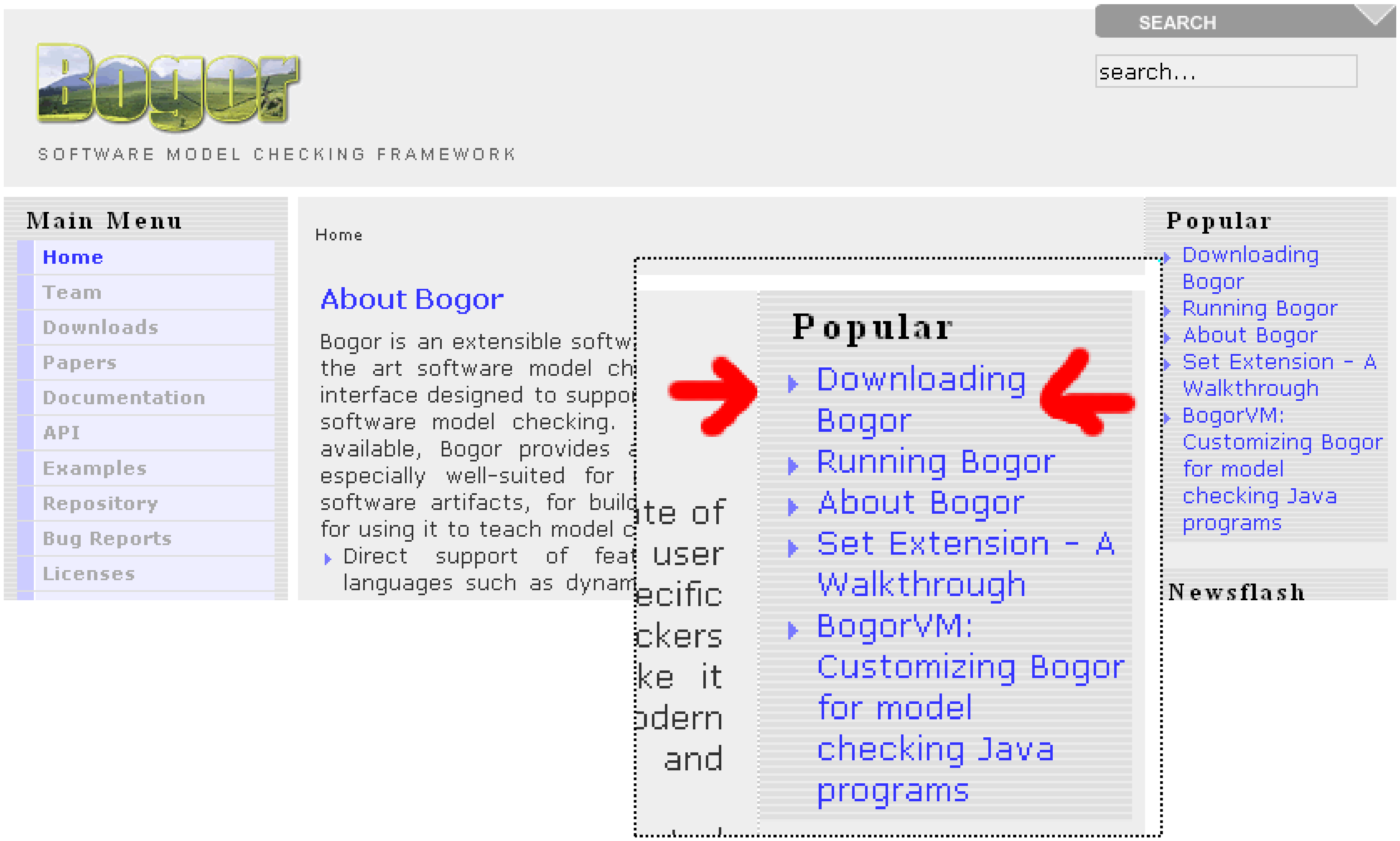
- Bogor is installed on the lab machines
- However, you might want to work from home, or you might just love Bogor

- Easiest to work with as an Eclipse Plugin
  - http://www.eclipse.org/
  - http://bogor.projects.cis.ksu.edu/

Installing Bogor

- Eclipse
  - Eclipse is, at its core, an open IDE for developing in a number of programming languages
  - You download a distribution based on what type of software you are developing, and a number of useful tools are included
  - For Bogor, this isn't so important, since we are just interested in the Eclipse IDE
  - I recommend: Eclipse IDE for Java Developers (85 MB) - unzip anywhere, it is self-contained

Installing Bogor



SEARCH

search...

SOFTWARE MODEL CHECKING FRAMEWORK

Main Menu

- Home
- Team
- Downloads
- Papers
- Documentation
- API
- Examples
- Repository
- Bug Reports
- Licenses

Home

About Bogor

Bogor is an extensible softw the art software model ch interface designed to suppo software model checking. available, Bogor provides a especially well-suited for software artifacts, for build for using it to teach model c

- Direct support of feat languages such as dynam

Popular

- Downloading Bogor
- Running Bogor
- About Bogor
- Set Extension - A Walkthrough
- BogorVM: Customizing Bogor for model checking Java programs

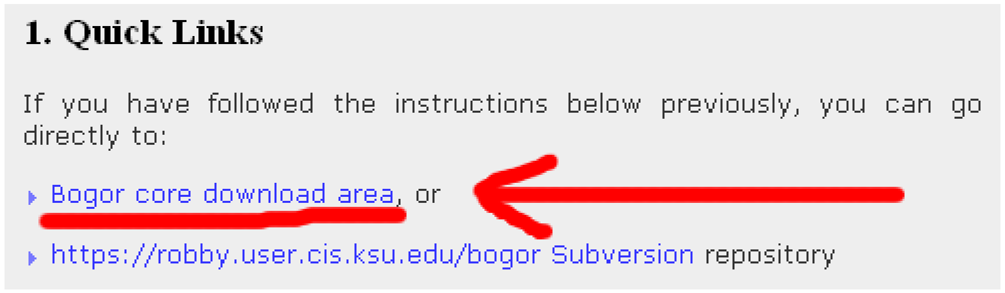
Newsflash

Popular

- Downloading Bogor
- Running Bogor
- About Bogor
- Set Extension - A Walkthrough
- BogorVM: Customizing Bogor for model checking Java programs

ate of user ecific ckers ke it odern and

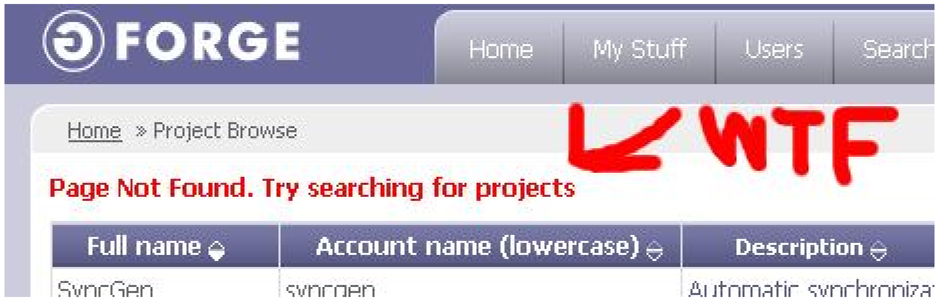
Installing Bogor



1. Quick Links

If you have followed the instructions below previously, you can go directly to:

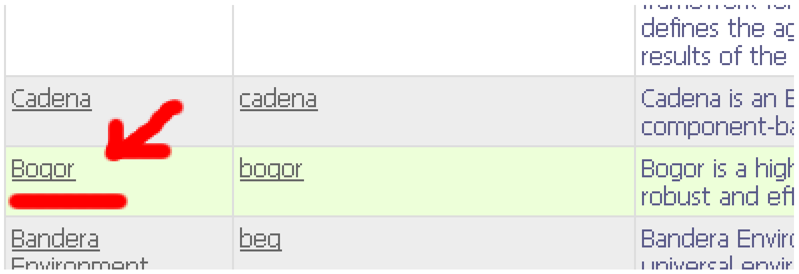
- Bogor core download area, or
- https://robby.user.cis.ksu.edu/bogor Subversion repository



Home » Project Browse

Page Not Found. Try searching for projects

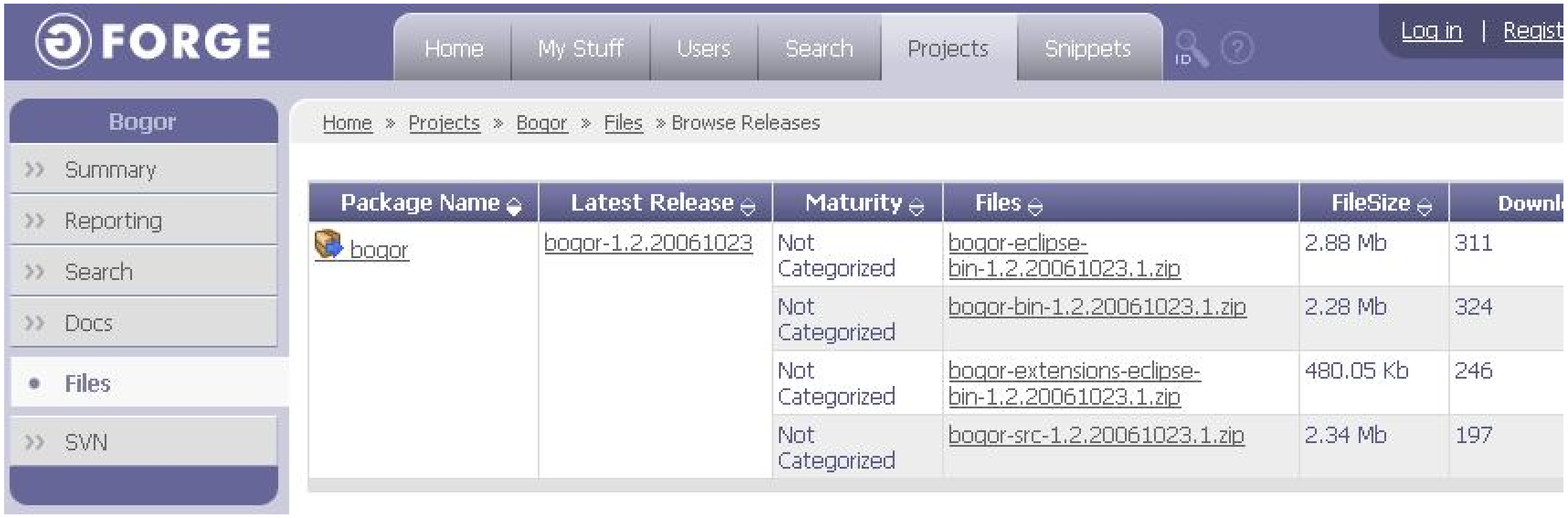
Full name	Account name (lowercase)	Description
SyncGen	syncgen	Automatic synchroniza



		defines the ag results of the
Cadena	cadena	Cadena is an E component-ba
Bogor	bogor	Bogor is a high robust and eff
Bandera Environment	beg	Bandera Enviro universal envir

Installing Bogor

- bogor-eclipse-bin-xxx.zip

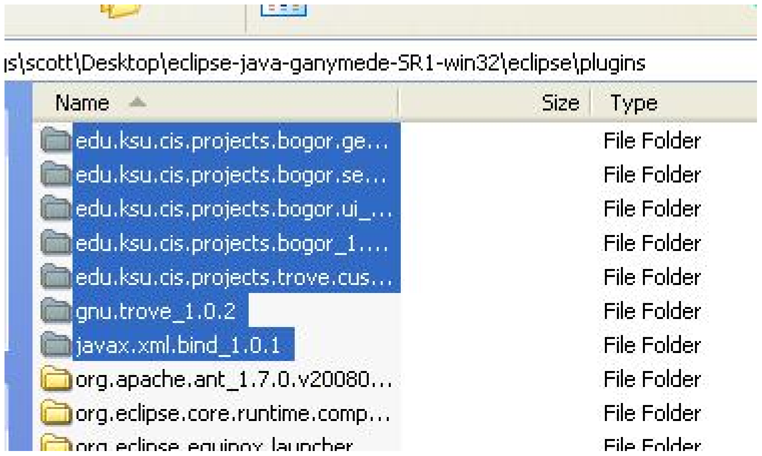


Home » Projects » Bogor » Files » Browse Releases

Package Name	Latest Release	Maturity	Files	FileSize	Downl
bogor	bogor-1.2.20061023	Not Categorized	bogor-eclipse-bin-1.2.20061023.1.zip	2.88 Mb	311
		Not Categorized	bogor-bin-1.2.20061023.1.zip	2.28 Mb	324
		Not Categorized	bogor-extensions-eclipse-bin-1.2.20061023.1.zip	480.05 Kb	246
		Not Categorized	bogor-src-1.2.20061023.1.zip	2.34 Mb	197

Installing Bogor

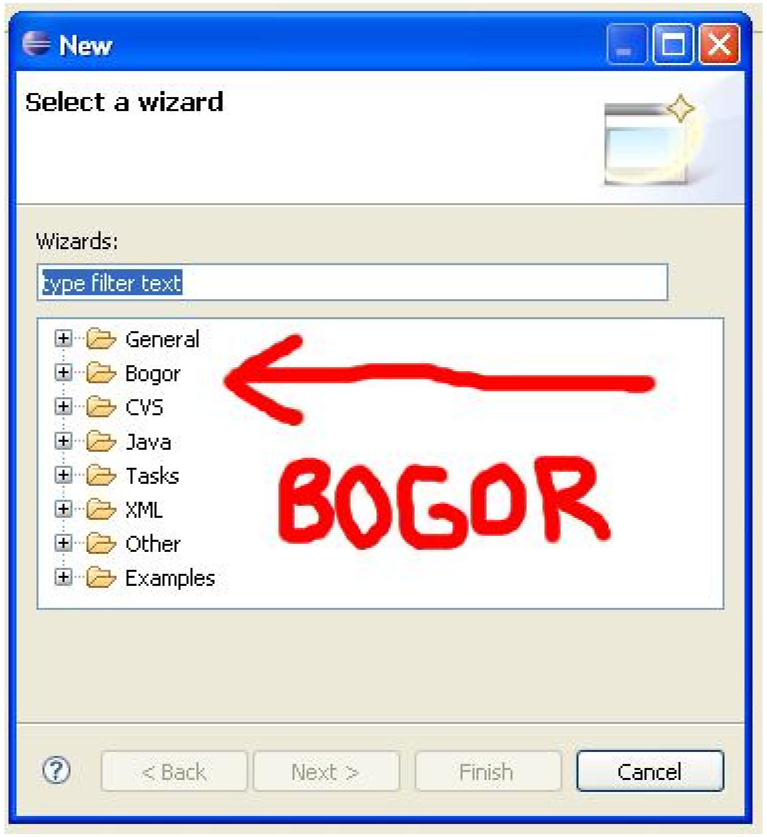
- Drag and Drop, FTW
  - bogor-eclipse-bin-xxx.zip should contain seven directories
    - Five start with edu.ksu.cis.projects, one starts with gnu.trove, and one starts with javax.xml.bind.
  - Copy all of these into the \eclipse\plugins directory of your new Eclipse installation.


s\scott\Desktop\eclipse-java-ganymede-SR1-win32\eclipse\plugins

Name	Size	Type
edu.ksu.cis.projects.bogor.ge...		File Folder
edu.ksu.cis.projects.bogor.se...		File Folder
edu.ksu.cis.projects.bogor.ui_...		File Folder
edu.ksu.cis.projects.bogor_1....		File Folder
edu.ksu.cis.projects.trove.cus...		File Folder
gnu.trove_1.0.2		File Folder
javax.xml.bind_1.0.1		File Folder
org.apache.ant_1.7.0.v20080...		File Folder
org.eclipse.core.runtime.comp...		File Folder
org.eclipse.equinox.launcher		File Folder

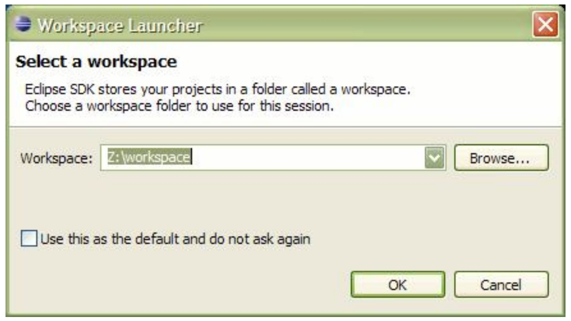
Installing Bogor

- How do I know it worked?
  - Start Eclipse, and go to File -> New -> Other



Starting Bogor

- In the labs
  - Start -> Programs -> Program Development -> Eclipse 3.3.0
- What's a "workspace"?
  - Eclipse stores information about your projects there, and you're fine to use the default..
  - .. but not in the labs, it seems. If you have problems, try:
    - z:\workspace



Select a workspace

Eclipse SDK stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

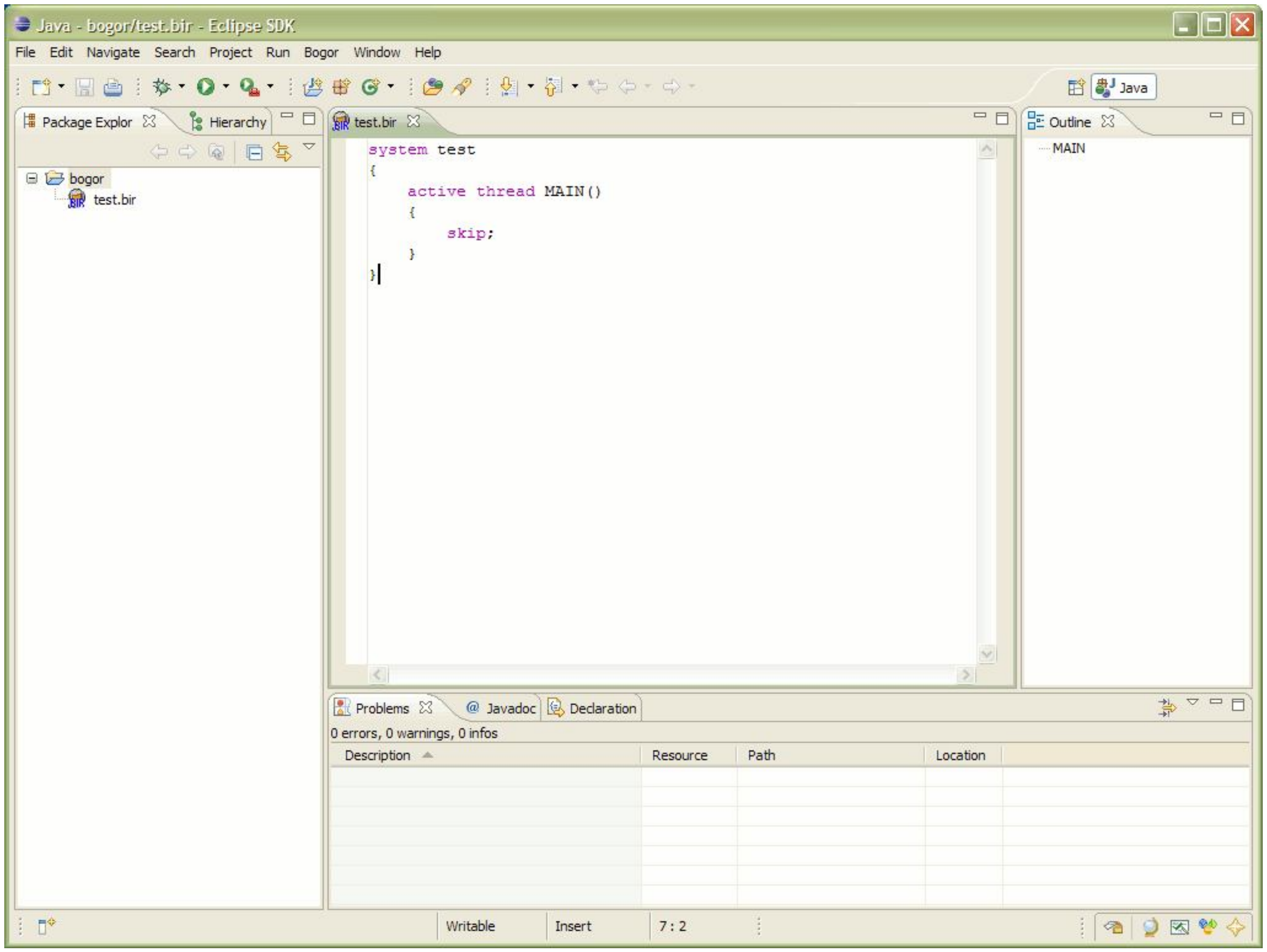
Workspace: Z:\workspace

☐ Use this as the default and do not ask again

Starting Bogor

- Create a project
  - File -> New -> Project
  - General -> Project
  - Project Name: whatever you'd like. bogor?
- Create a BIR Model in which to write code
  - File -> New -> Other
  - Bogor -> BIR Model
  - File Name: your_file.bir
    - (Filename must end with .bir, otherwise it's up to you)

Starting Bogor



```
system test
{
    active thread MAIN()
    {
        skip;
    }
}
```

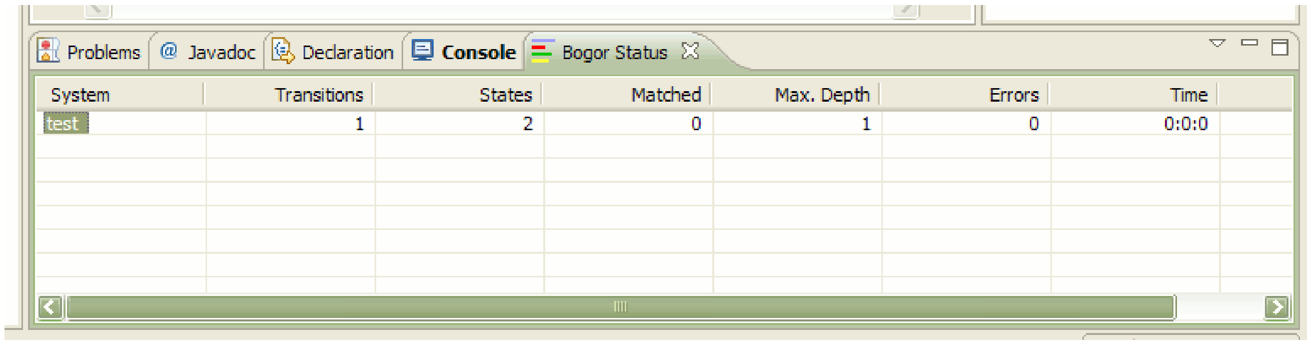
0 errors, 0 warnings, 0 infos

Starting Bogor

- How do you actually run Bogor?
  - Bogor -> Model Check
    - (Or you can right-click in the work area, and choose Model Check)
  - Choose "Config 0: Default Configuration"
    - If you don't explicitly choose it, nothing is selected by default. This sounds pedantic, but it can be confusing when you choose OK, and nothing runs. Gah!

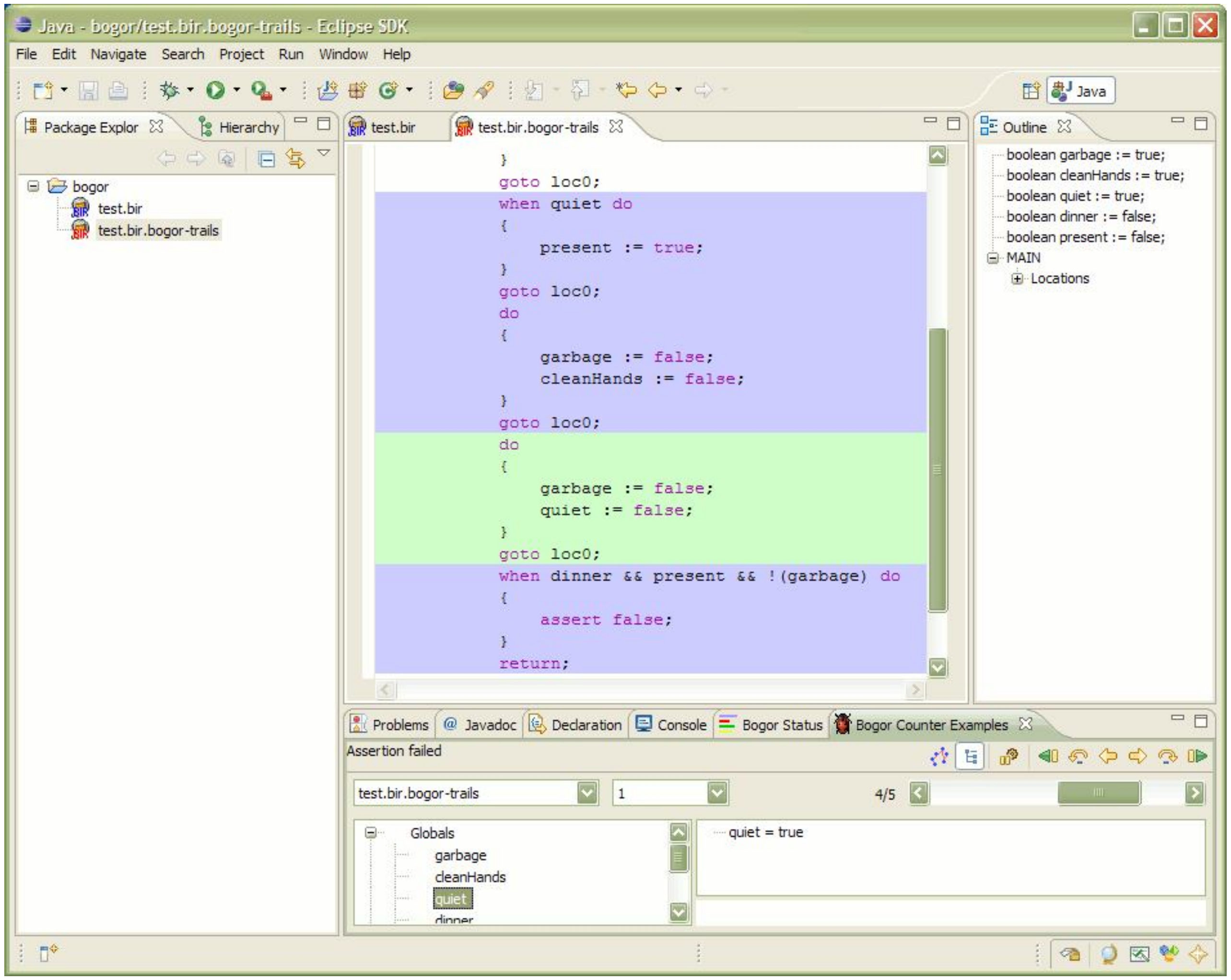
Starting Bogor

- What am I looking at?
  - Bogor writes a trail file with the extension bogor-trails if any errors have been found.
  - The trail file contains schedule information and state transitions that lead to the errors.
  - You can open the counter-example display by double-clicking the trail file.



System	Transitions	States	Matched	Max. Depth	Errors	Time
test	1	2	0	1	0	0:0:0

Starting Bogor



```
        }
        goto loc0;
        when quiet do
        {
            present := true;
        }
        goto loc0;
        do
        {
            garbage := false;
            cleanHands := false;
        }
        goto loc0;
        do
        {
            garbage := false;
            quiet := false;
        }
        goto loc0;
        when dinner && present && !(garbage) do
        {
            assert false;
        }
        return;
```

```
boolean garbage := true;
boolean cleanHands := true;
boolean quiet := true;
boolean dinner := false;
boolean present := false;
MAIN
  Locations
```

Assertion failed

test.bir.bogor-trails   1   4/5

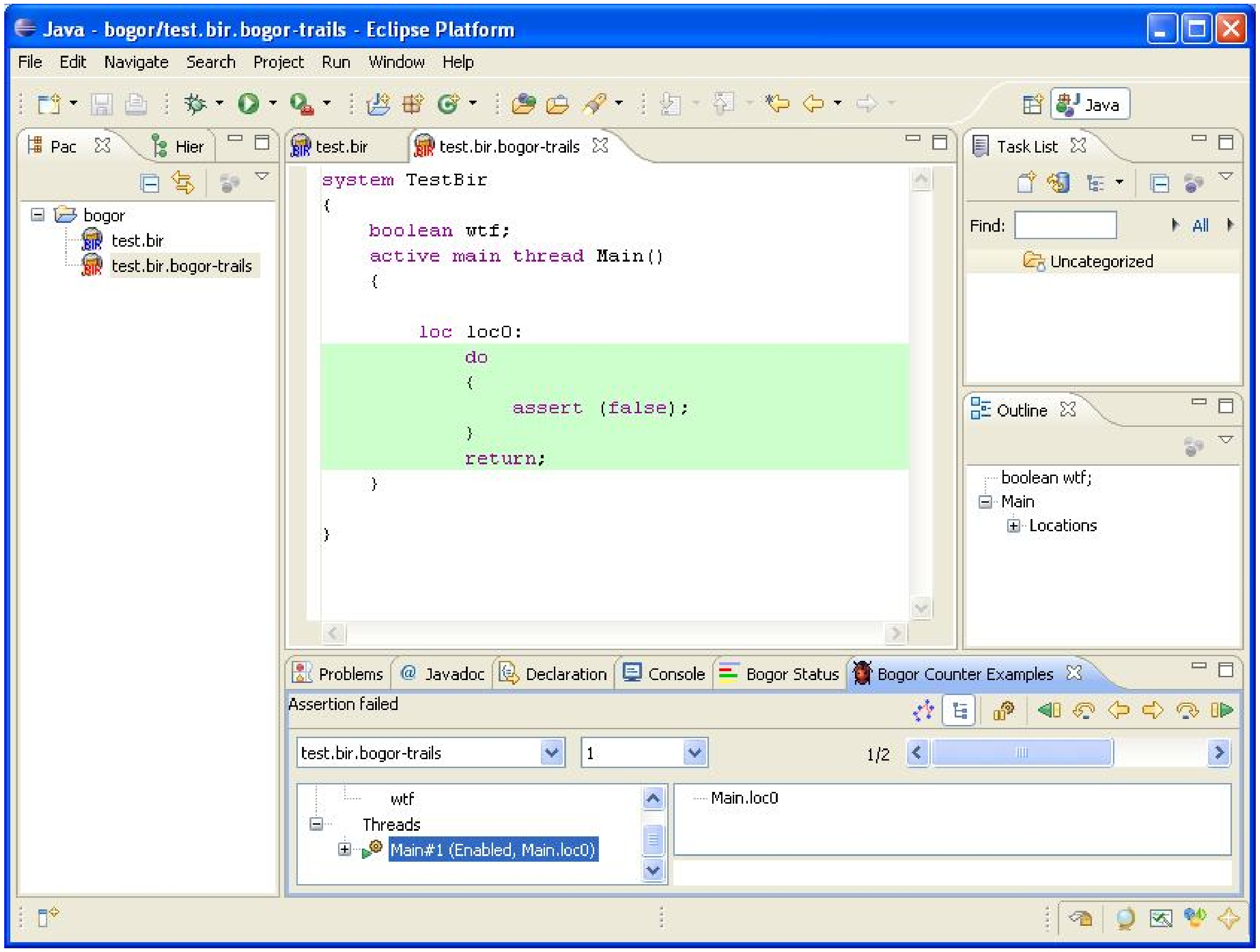
```
Globals
  garbage
  cleanHands
  quiet
  dinner
```

quiet = true

General Usage and Tips

- Bogor Trails
  - Double-click on the bogor-trails file, if it is generated after an error occurs.
  - You can examine each error trail, at each step of execution, and can observe the values that caused the error to occur.
  - These trails can get quite large in some examples!

General Usage and Tips



```
system TestBir
{
    boolean wtf;
    active main thread Main()
    {

        loc loc0:
            do
            {
                assert (false);
            }
            return;
    }

}
```

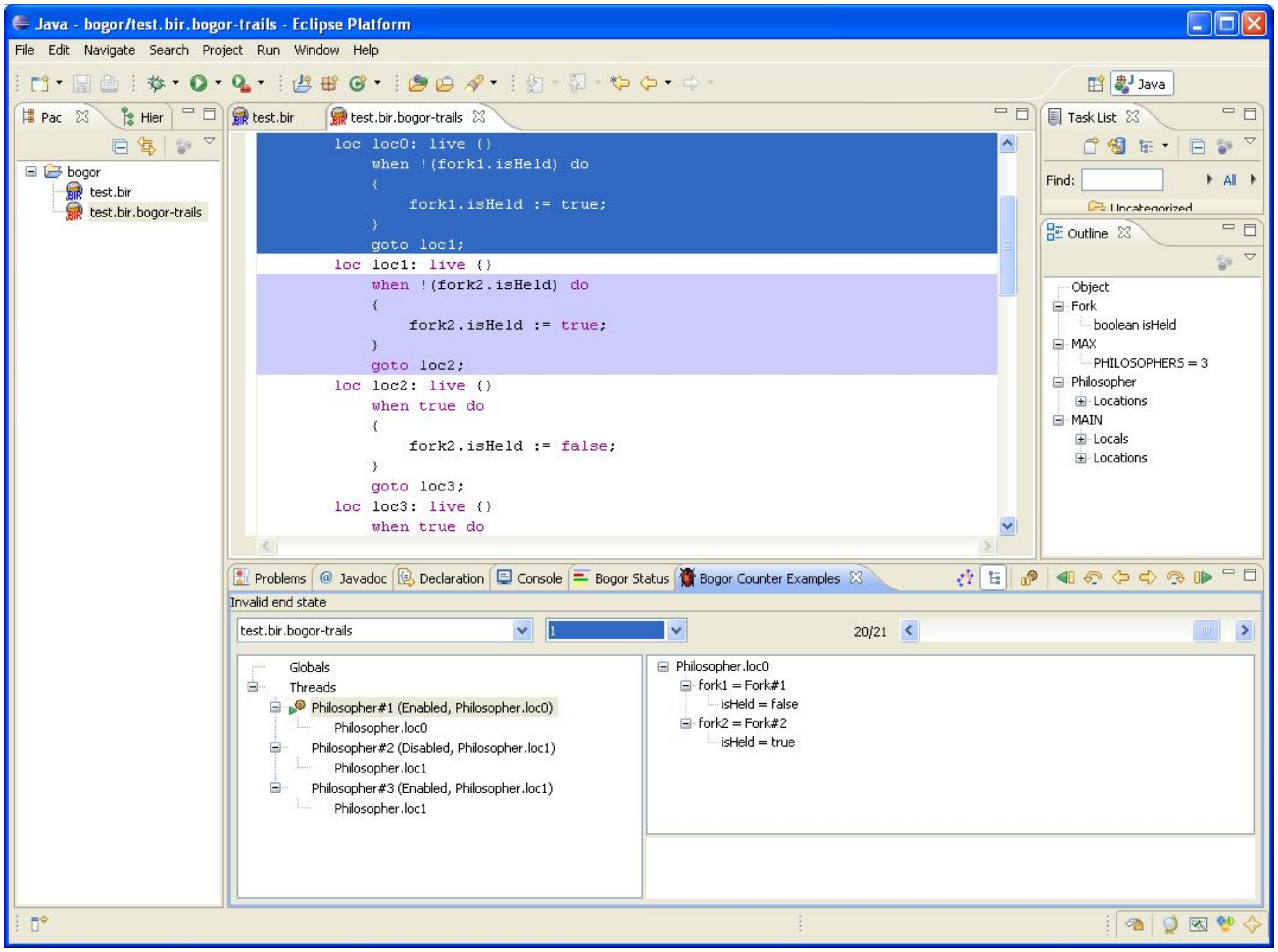
Assertion failed

test.bir.bogor-trails | 1 | 1/2

```
wtf
Threads
  Main#1 (Enabled, Main.loc0)
```

Main.loc0

General Usage and Tips



```
bogor
    test.bir
    test.bir.bogor-trails
```

```
        loc loc0: live {}
            when !(fork1.isHeld) do
            {
                fork1.isHeld := true;
            }
            goto loc1;
        loc loc1: live {}
            when !(fork2.isHeld) do
            {
                fork2.isHeld := true;
            }
            goto loc2;
        loc loc2: live {}
            when true do
            {
                fork2.isHeld := false;
            }
            goto loc3;
        loc loc3: live {}
            when true do
```

```
Object
Fork
    boolean isHeld
MAX
    PHILOSOPHERS = 3
Philosopher
    Locations
MAIN
    Locals
    Locations
```

Invalid end state

test.bir.bogor-trails   1   20/21

```
Globals
Threads
    Philosopher#1 (Enabled, Philosopher.loc0)
        Philosopher.loc0
    Philosopher#2 (Disabled, Philosopher.loc1)
        Philosopher.loc1
    Philosopher#3 (Enabled, Philosopher.loc1)
        Philosopher.loc1
```

```
Philosopher.loc0
    fork1 = Fork#1
        isHeld = false
    fork2 = Fork#2
        isHeld = true
```

General Usage and Tips

- Understand why paths are taken
- Watch assignments occur
- Remember how model checking can be used to solve these problems
  - For Q1/Q2, you actually want the model to "fail" when you have reached your goal

Advice for Assignment 1

- Q1
  - How are you going to represent the position of each element that crosses the river?
    - Up to you. In this case, you only really need to know which side of the shore each element is on, so a boolean or int would make sense
  - How can you prevent illegal positions from being considered?
    - Use assume(b), where b is some invalid state that we do not want to pursue

Advice for Assignment 1

```
system TestBir {
  int n;

  main thread Main() {
    loc loc0:
      do {
        n := n + 1;
      } goto loc1;
    loc loc1:
      do {
      } goto loc0;
  }
}
```

Yikes!

```
system TestBir {
  int n;

  main thread Main() {
    loc loc0:
      do {
        n := n + 1;
      } goto loc1;
    loc loc1:
      do {
        assume n < 100;
      } goto loc0;
  }
}
```

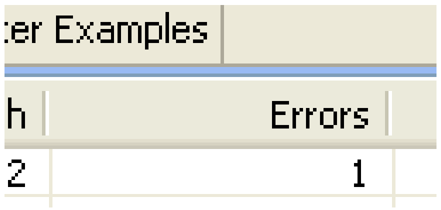
Advice for Assignment 1

- Q1
  - An assertion can demonstrate that you've reached the goal state

I claim that this model will allow n to reach a value greater than 10. How can I use Bogor to prove it? By asserting that in loc1, at some point n will not be less than or equal to 10.

```
system TestBir {
  int n;

  main thread Main() {
    loc loc0:
      do {
        n := n + 1;
      } goto loc1;
    loc loc1:
      do {
        assert n <= 10;
      } goto loc0;
  }
}
```



h	Errors
2	1

Advice for Assignment 1

- Q2
  - Many of the same rules apply, but you might need to find a different representation for the position of your elements
    - You have three shores now, so if you used one boolean previously for position, it won't be enough on its own for three positions
  - Remember how atomicity works in a location, and use the invisible keyword if necessary

Advice for Assignment 1

- Q2
  - Each farmer should be represented by its own thread
  - Otherwise, the same rules from Q1 apply!
    - Use assume to prune invalid subtrees
    - Use assert to identify the goal state
    - Don't let farmers make invalid moves, but remember that they're allowed to travel alone, as long as they don't leave two incompatible items alone on the same shore

Advice for Assignment 1

- Q3
  - Q3a asks for a brief explanation on why the simulation is insufficient to determine if the property always holds no matter how long the simulation is run.
  - Be brief!

Advice for Assignment 1

- Q3
  - Each worker in the cooperative should be represented by their own thread in memory, but the thread definition should allow for more than three workers

```
system SleeplessCode
{
  const C { N = 2; }
  // ...
  thread Worker(int id) {
    // ...
  }
```

```
active thread MAIN()
{
  int counter;
  // ...
  loc loc1:
    when counter < C.N do {
      counter := counter + 1;
      start Worker(counter);
    } goto loc1;
    when counter == C.N do {} return;
```

Advice for Assignment 1

- Q3
  - You'll need to store the rank of each worker, and the last worker to have reached a certain rank. These values will need to be observed by each Worker thread..

```
system SleeplessCode
{
  // ...
  int[] ranks;
  int[] last_promoted_at_rank;
```

```
active thread MAIN()
{
  loc loc0:
  do {
    ranks := new int[C.N + 1];
    last_promoted_at_rank := new int[C.N + 1];
  } goto loc1;
```

Advice for Assignment 1

- Q3
  - You should be able to convince yourself whether or not the properties hold
    - Step through the code, and if necessary, insert assertions to see how things progress
    - Adding assertions can be a great way to identify why your model is doing something you think it shouldn't
  - Remember how the rules of atomicity work, and remember at which points each thread can take over execution
  - Give yourself time on Q3 - it can be tricky!

Advice for Assignment 1

- If something seems unclear, explain why you made a decision
  - It's easier to read commented code than obfuscated code, and if you leave reasoning for your decisions, your intentions are clearer

Questions?

- Hope this is helpful!