Concurrency

EECS 4315

www.eecs.yorku.ca/course/4315/

The readers-writers problem

The readers and writers problem, due to Courtois, Heymans and Parnas, is a classical concurrency problem. It models access to a database. There are many competing threads wishing to read from and write to the database. It is acceptable to have multiple threads reading at the same time, but if one thread is writing then no other thread may either read or write. A thread can only write if no thread is reading.

Database class

```
public class Database {
  private boolean writing;
  private int readers;
  ...
  public void read() {
    this.beginRead();
    // read
    this.endRead();
  }
}
```

Add `assert !this.writing` in the `read` method where the database is read. If the assertion fails, an exception is thrown. JPF detects exceptions that are thrown and not caught.

Configuring the report

```
sourcepath=<path to ReadersAndWriters.java>

report.class=gov.nasa.jpf.report.Reporter
report.publisher=console

report.console.property_violation=error,trace
report.console.show_steps=true
report.console.show_method=true
report.console.show_code=true
```

Report

```
JavaPathfinder core system v8.0 (rev 32+) - (C) 2005-2014 U
========================================================== sea
========================================================== erro
gov.nasa.jpf.vm.NoUncaughtExceptionsProperty
java.lang.AssertionError
at concurrency.Database.read(concurrency/Database.java:30)
at concurrency.Reader.run(concurrency/Reader.java:25)
========================================================== trac
---------------------------------------------------------- tran
gov.nasa.jpf.vm.choice.ThreadChoiceFromSet {id:"ROOT" ,1/1,
  java.lang.Boolean.[<clinit>]
    invokestatic java.lang.Boolean.<clinit>()V
  java.lang.Boolean.<clinit>()V
    new java.lang.Boolean@bc
    dup
    iconst_1
    invokespecial java.lang.Boolean.<init>(Z)V
  java.lang.Boolean.<init>(Z)V
```

Installing jpf-shell

The extension jpf-shell can be downloaded from the Mercurial repository `http://babelfish.arc.nasa.gov/hg/jpf/jpf-shell`.

How to install an extension of JPF is described in the Section 1.9 of the notes.

Using jpf-shell

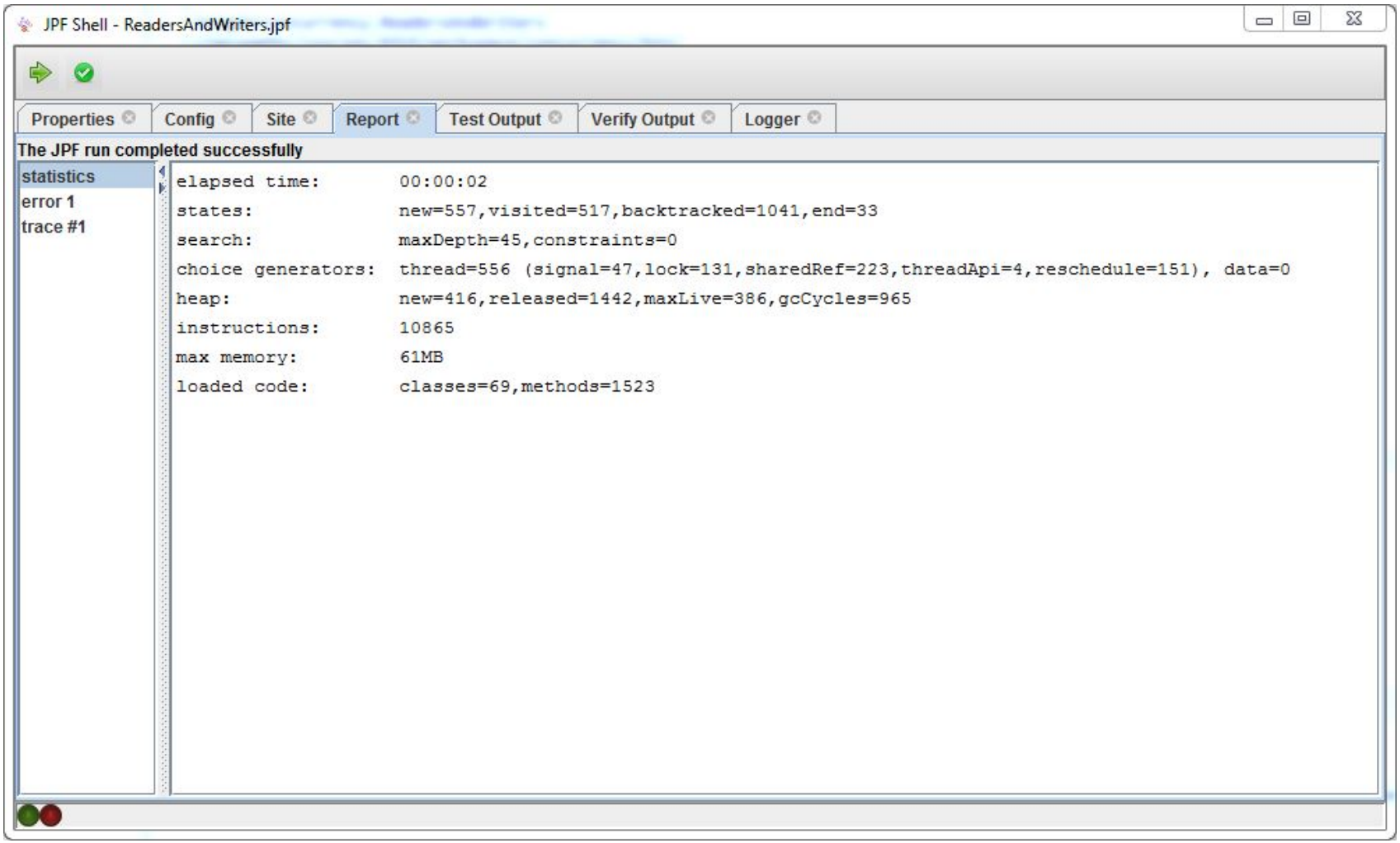
Site properties file:

```
# JPF site configuration

jpf-core=<path to jpf-core>
jpf-shell=<path to jpf-shell>

extensions=${jpf-core}
```

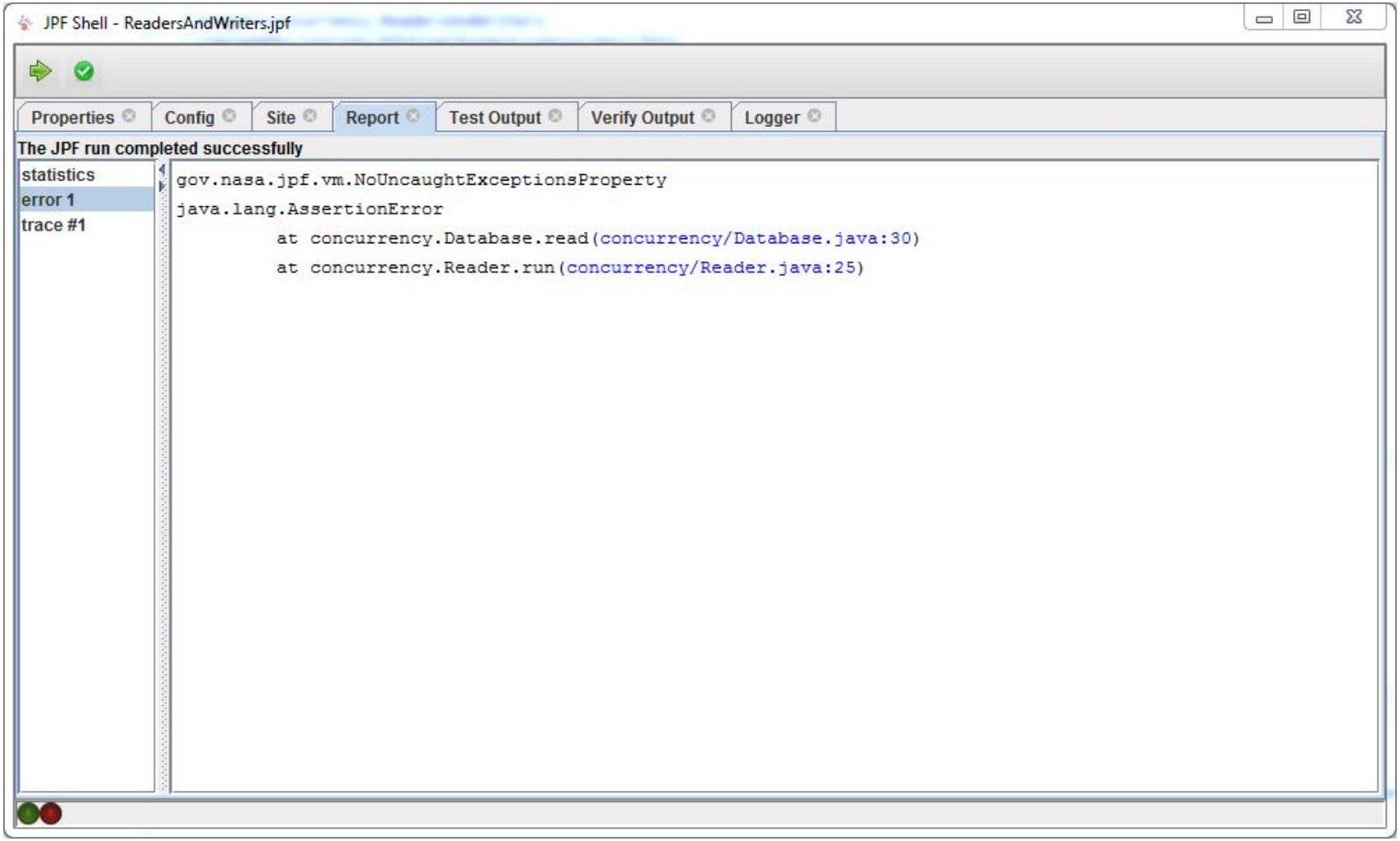
Configuration file:

```
target=ReadersAndWriters
classpath=<path to ReadersAndWriters.class>
sourcepath=<path to ReadersAndWriters.java>

@using=jpf-shell
shell=.shell.basicshell.BasicShell
```

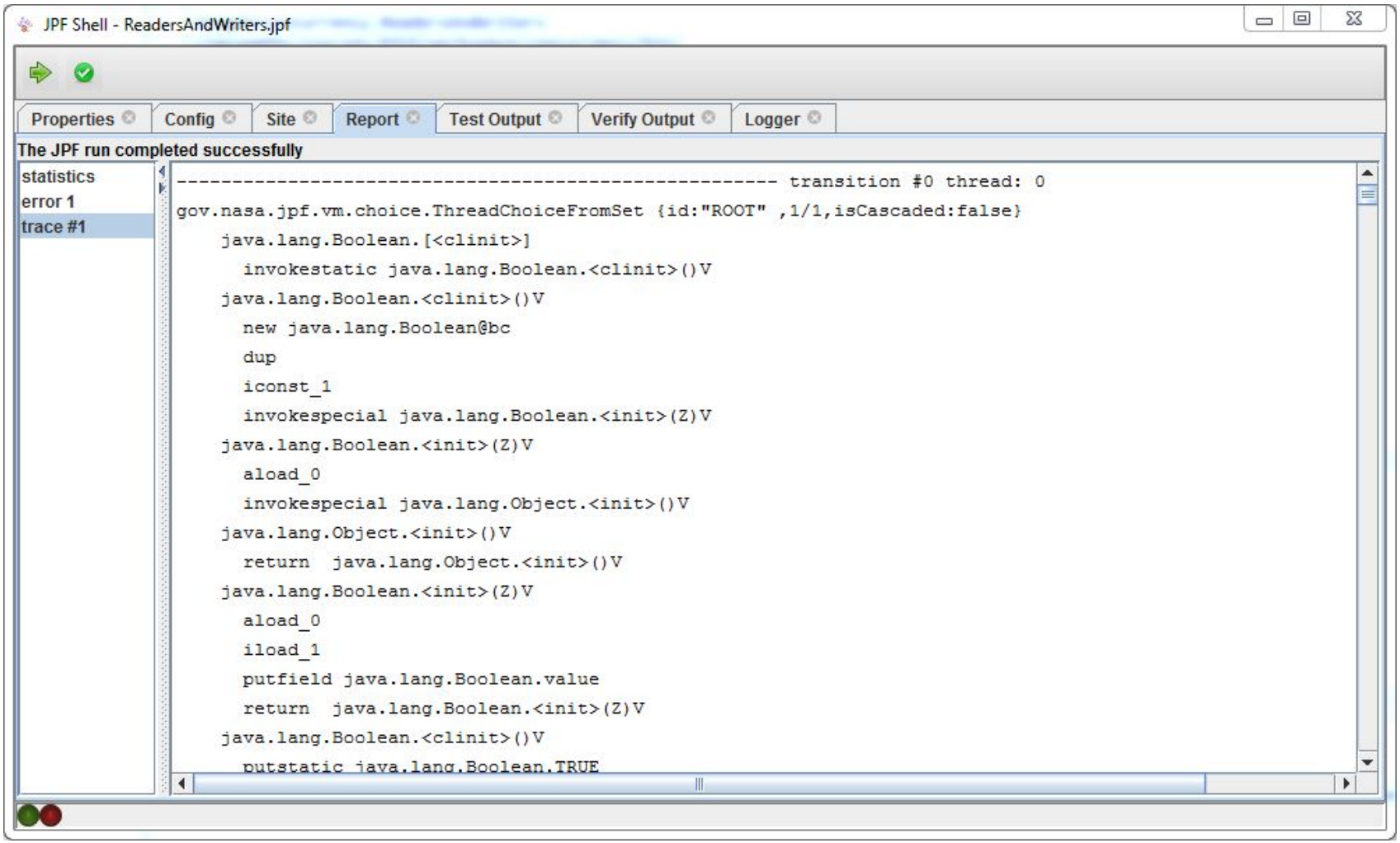
Report of jpf-shell



Report of jpf-shell



Report of jpf-shell



Cyrille Artho and Qiyi Tang. JPF Visual. Presented at the JPF Workshop, Urbana-Champaign, IL, USA, November 2017.

Installing jpf-shell

The extension jpf-visual can be downloaded from the BitBucket repository `https://bitbucket.org/qiyitang71/jpf-visual`.

How to install an extension of JPF is described in the Section 1.9 of the notes.

Using jpf-visual

Site properties file:

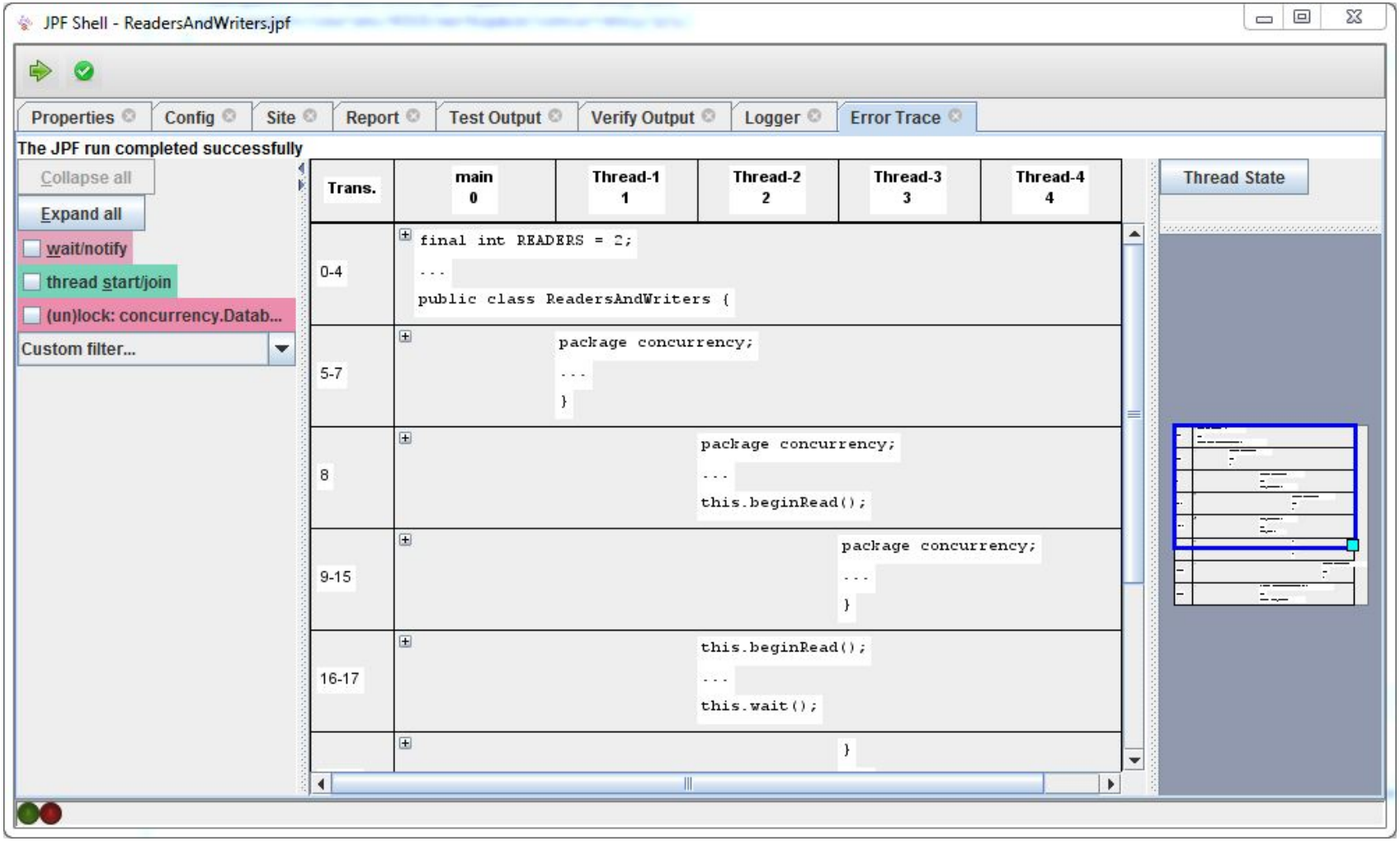
```
# JPF site configuration

jpf-core=<path to jpf-core>
jpf-shell=<path to jpf-shell>
jpf-visual=<path to jpf-visual>

extensions=${jpf-core}
```

Using jpf-visual

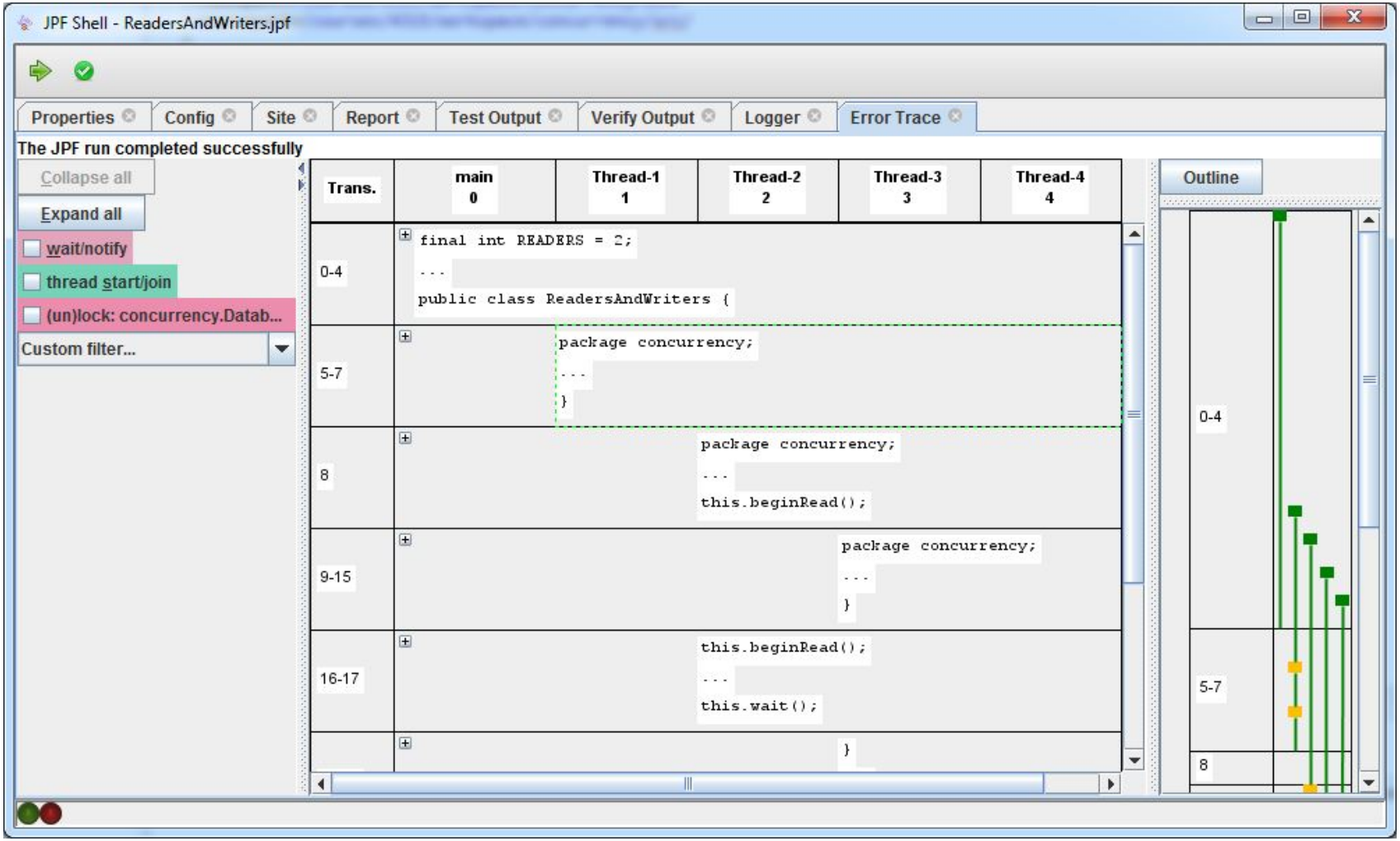
Configuration file:

```
target=ReadersAndWriters
classpath=<path to ReadersAndWriters.class>
sourcepath=<path to ReadersAndWriters.java>

@using=jpf-visual
shell=.shell.basicshell.BasicShell
shell.panels+=,errorTrace
shell.panels.errorTrace=ErrorTracePanel
report.publisher+=,errorTracePrinter
report.errorTracePrinter.class=ErrorTracePrinter
report.errorTracePrinter.property_violation=trace
```

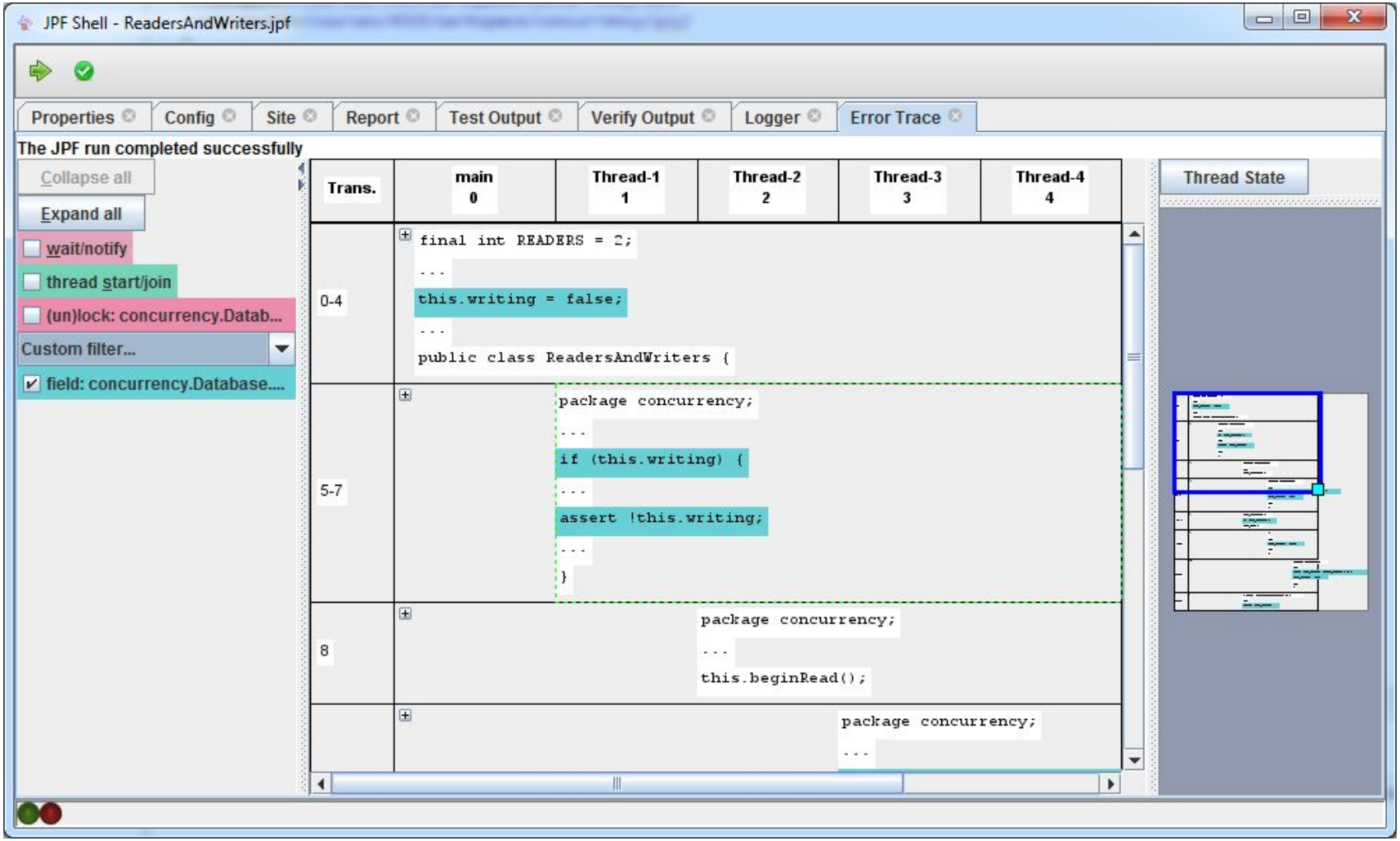
Report of jpf-visual



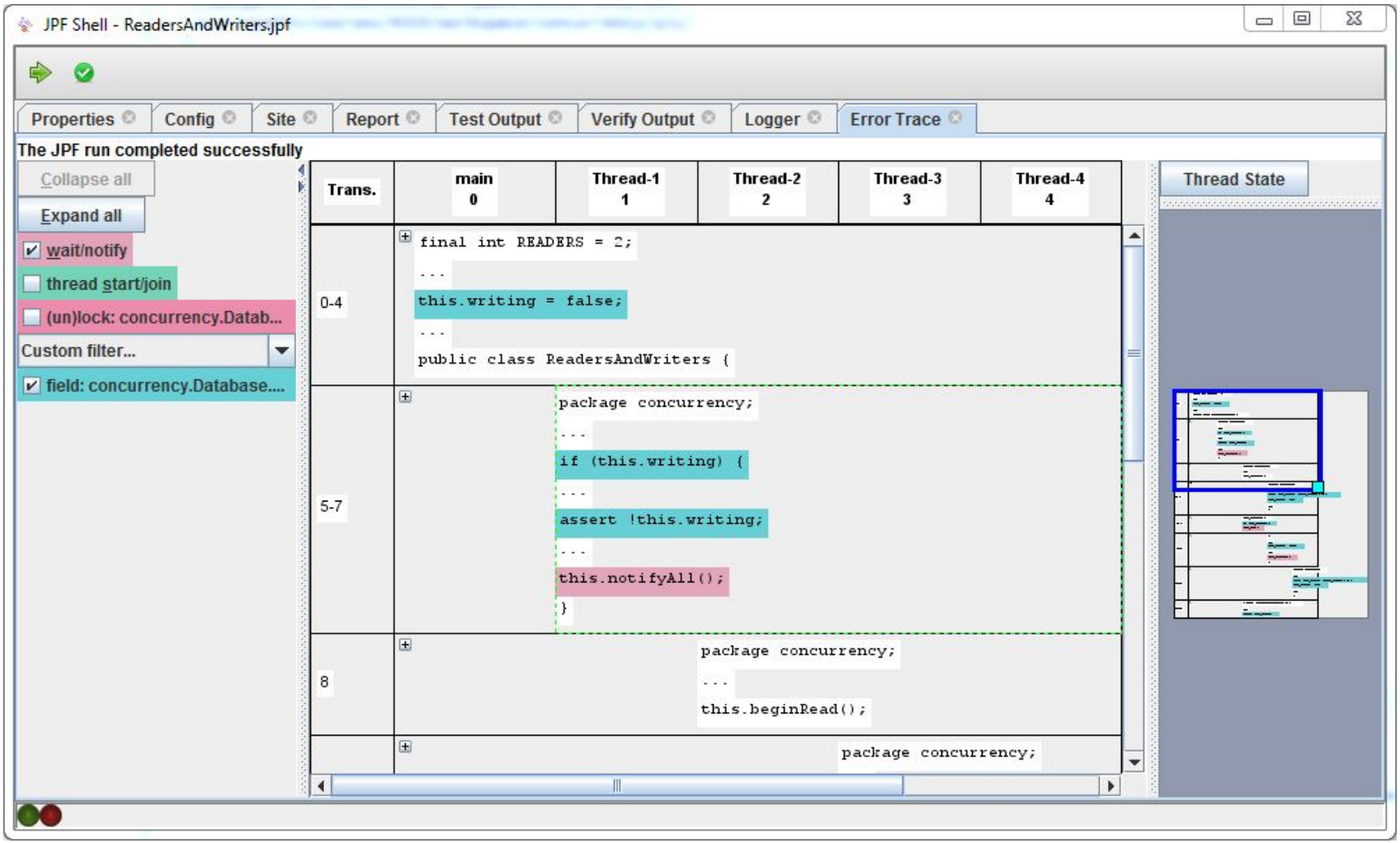
Report of jpf-visual



Report of jpf-visual



Report of jpf-visual



Concurrent stack

- `Popper` class
- `Pusher` class
- `Stack` class
- `Main` class

Dining savages problem

A tribe of savages eats communal dinners from a large pot that can hold M servings of stewed missionary. When savages want to eat, they help themselves from the pot, unless it is empty. If the pot is empty, the savage wakes up the cook and then waits until the cook has refilled the pot.