# GPars - Groovy Parallel Systems

Jon Kerridge

Version 1.0, 2015-10-29

# Table of Contents

Testing Parallel Systems: First Steps . . . . . . . . . . 1

Testing Hello World . . . . . . . . . . 1

# Testing Parallel Systems: First Steps

**JUnit** testing has become a widely accepted way of testing **Java** classes and there is a great deal of software support for this approach. In the previous chapter, examples and exercises were introduced whereby the user had to ascertain for themself that the systems worked in the expected manner. This was achieved by looking at printed output.  This may be a satisfactory approach for small example systems but is not appropriate for systems that are to be used in an every day context.

In this chapter, the use of **JUnit** testing is introduced by using examples taken from earlier chapters. This will demonstrate that it is possible to use this approach and give a general architecture for testing parallel systems. The key to **JUnit** testing is that we test one or more assertions concerning the underlying implementation.

In the parallel situation, we have to identify a source of inputs that can be compared to the subsequent outputs for the assertion testing.

## Testing Hello World

The testing of the **ProduceHW** and **ConsumeHW** processes (see Chapter 2) demonstrates that from the outset, testing has to be considered at the time processes are designed and cannot be retrospectively added. To this end, properties are required that can be accessed once a process has terminated.

These properties can then become components in any assertion. In this very simple case the **ProduceHW** process needs no alteration.

Want to read more of this chapter? Download this chapter’s PDF here.