

# The Craftsman: 19

## SMCRemote Part IX

### Tolerance

Robert C. Martin  
16 Oct, 2003

*...Continued from last month. You can download last month's code from:  
[www.objectmentor.com/resources/articles/CraftsmanCode/Craftsman\\_18\\_Slow\\_And\\_Steady.zip](http://www.objectmentor.com/resources/articles/CraftsmanCode/Craftsman_18_Slow_And_Steady.zip)*

---

Jean took me down the turbo to the journeyman's lounge at .36g in the gamma arm. She said the low-g did wonders for her joints. When we got there I noticed Jerry, Jasmine, and a few other journeymen clustered at a table. Jean noticed them too and started walking over to them. I followed reluctantly.

"Hello my dears!" she exclaimed as though she hadn't seen them in weeks. They all greeted her with similar warmth. Then she excused herself and trotted over to a vacant head, leaving me to face my two previous mentors alone.

"Hello Jerry; Ms. J. How are you?"

Jerry looked perplexed. "What's this Ms. J. stuff about?"

Jasmine impatiently brushed this aside and said: "Drop the Ms. J stuff hotshot. We've had enough of that." Her eyes were as disarming as ever, and I found myself unable to answer her intelligently.

A coffee servitor rolled by. Grateful for the distraction I grabbed a cup and then faced my mentors again.

"How's Jean treating you hotshot?" Jasmine asked.

Jerry leaped in with: "Yeah, how'd you score a gig with *her*?"

After barely being able to say more than three words at a time for the whole morning with Jean, my frustrations, came pouring out.

"I didn't score anything, she just showed up this morning. Frankly working with her is a little frustrating. She's always calling me "Dear"; and she talks a *lot* and takes a *lot* of breaks. We barely get anything done. I'm not sure I want to keep working with her."

Jasmine and Jerry looked at me, then at each other, then starting laughing. Jerry settled down quickly; but Jasmine couldn't seem to gain control of her self. Every time she looked at me she spurted out new laughs that sounded like a sea lion calling to its mate.

"What?" I said.

Jerry took me aside while Jasmine continued to emit barks of laughter like shrill hiccoughs. "Alphonse, you don't understand how lucky you are. Any of us here would give our eyeteeth to work with Jean. I know she's a little quirky; but don't let the grandmother facade fool you. She's one of the best there is, and you'll learn a lot from her."

I was dumbfounded; but I couldn't respond because Jean chose that moment to toddle back from the head.

"My Goodness, I feel much better now."

This was more information than I wanted, so I changed the subject by offering to call the coffee servitor.

"Oh, my no! Dear. Why that'd just bring me right back here in ten or twenty minutes. No, I think we ought to go back up to our uncomfortable .6g lab and keep working on SMCRemote, don't you? We've got a lot of ground yet to cover today, and we certainly aren't getting it done down here, are we? ...What is Jasmine making that terrible sound for? Sounds like some animal dying. Jasmine, drink some water dear...

We took the turbo back out to the rim and walked back to the lab. The nearly doubled g slowed Jean down quite a bit. She was much spryer down in the lounge.

After we got situated back at our workstation, she said: "Now then Alphonse dear, I think we'd better see if we can execute that command we just built. What do you think?"

I had been wondering about how we were going to do this, so I agreed. "Er, yes."

"Good dear. Now, since the command we are about to execute invokes the SMC compiler, I think the first thing we need to do is create a simple source file for that compiler to read. So let's write a test case that creates this file, then invokes the compiler, and then checks to see that the compiler has created the proper output files."

She had pulled out her knitting again, and showed no sign that she wanted to touch the keyboard, so I took it and started to type:

```
public void testExecuteCommand() throws Exception {
    File sourceFile = new File("simpleSourceFile.sm");
    PrintWriter pw = new PrintWriter(new FileWriter(sourceFile));
    pw.println("
}
```

I didn't know how to continue, so I looked expectantly at her.

"That's fine dear. Here, I'll type in the SMC syntax for you."

```
public void testExecuteCommand() throws Exception {
    File sourceFile = new File("simpleSourceFile.sm");
    PrintWriter pw = new PrintWriter(new FileWriter(sourceFile));
    pw.println("Context C");
    pw.println("FSMName F");
    pw.println("Initial I");
    pw.println("{I{E I A}}");
    pw.close();
}
```

"What does that mean, Jean?"

"Well, dear, for our purposes it means that the compiler will create a file named F.java. That's all you really need to know about it for now. However, once you get back to your quarters it would behoove you to look up the SMC document and read it. I wrote it quite a few years ago dear, and I still think it's one of my better documents. It's called: 'Care and Feeding of the State Map Compiler'<sup>1</sup>. Now then, let's see if we can execute that compile command."

I wasn't quite sure how to execute the command; but I'd learned from Jerry and Jasmine that it's often better just to write the calls that express my intention. So I continued to type:

```
public void testExecuteCommand() throws Exception {
    File sourceFile = new File("simpleSourceFile.sm");
    PrintWriter pw = new PrintWriter(new FileWriter(sourceFile));
    pw.println("Context C");
    pw.println("FSMName F");
    pw.println("Initial I");
}
```

---

<sup>1</sup> You can get this document from: <http://www.objectmentor.com/resources/downloads/bin/smcJava.zip>

```

pw.println("{I{E I A}}");
pw.close();

String command = SMCRemoteServer.buildCommandLine("simpleSourceFile.sm");
assertEquals(true, SMCRemoteServer.executeCommand(command));

File outputFile = new File("F.java");
assertTrue(outputFile.exists());
assertTrue(outputFile.delete());
assertTrue(sourceFile.delete());
}

```

"Very nicely done, Alphonse. I think that captures the test case quite admirably. We haven't written `executeCommand` yet, but we can certainly express how we want it to be called, can't we. Now, let's stub that function out and watch this test case fail. It's always fun to watch them fail, don't you think?"

So I clicked on `executeCommand` and selected "Create Method", and my IDE created just the method stub I wanted. And, sure enough, the test failed.

```

public class SMCRemoteServer {
    ...
    public static boolean executeCommand(String command) {
        return false;
    }
}

```

"All right dear, now let's make that test pass."

Jean was sounding tired again. I knew she'd want another break soon. It was almost lunchtime, so I was hoping she could hold out until then. I quickly searched the javadocs to find out how to execute a command. Jean saw what I was doing and said:

"It's in the `Runtime` class, dear. There's a method named `exec`."

Sure enough, it was right where she said. So I typed the code that I thought would work.

```

public static boolean executeCommand(String command) {
    Runtime rt = Runtime.getRuntime();
    try {
        rt.exec(command);
        return true;
    } catch (IOException e) {
        return false;
    }
}

```

But when I ran the test it failed to find the output file. I looked in the directory and, sure enough, `F.java` was not there.

"Why is this failing, Jean?"

She looked up from her knitting and said: "You didn't wait for the process to finish, dear. When you execute a command it creates a new process that runs concurrently with yours. You have to wait for it to complete before you exit from `executeCommand`."

I examined the JavaDoc for `Runtime.exec`, and found that it returned a `Process` object. I also found that you could wait for the `Process` object to complete, and that you could query it for its exit status. So I made the following changes.

```

public static boolean executeCommand(String command) {
    Runtime rt = Runtime.getRuntime();
    try {
        Process p = rt.exec(command);

```

```
        p.waitFor();  
        return p.exitValue() == 0;  
    } catch (Exception e) {  
        return false;  
    }  
}
```

This time the test passed.

"That wasn't that hard." I said.

"Of course not dear. All we are doing is running a command. Of course it'll get a bit more complicated when we have to capture the messages that the compiler usually prints on the console. We'll have to bind to the standard output, and standard error of the `Process`. But let's do that after lunch dear, I'm starting to get hungry, aren't you?"

I sighed and stood up. It still seemed to me that we were going slowly. Though in hindsight I see that in my first half day with Jean we had finished up the client, and gotten the server to run a compile. We hadn't spent any time debugging. Perhaps we were moving faster than I thought.

Jean put the knitterbot back in her bag and held out a sweater. "Hear, dear, try this on."

If nothing else, this "gig" with Jean was going to teach me a new kind of tolerance.

*To be continued...*

---

*The code that Alphonse and Jean finished can be retrieved from:*

*[www.objectmentor.com/resources/articles/CraftsmanCode/Craftsman\\_19\\_Tolerance.zip](http://www.objectmentor.com/resources/articles/CraftsmanCode/Craftsman_19_Tolerance.zip)*