

FXJDI Test Specification

1. Project Description

The specification aims at describing, developing, and indicating progress of Tests to cover the high level functionality exposed by the FX JDI public API like

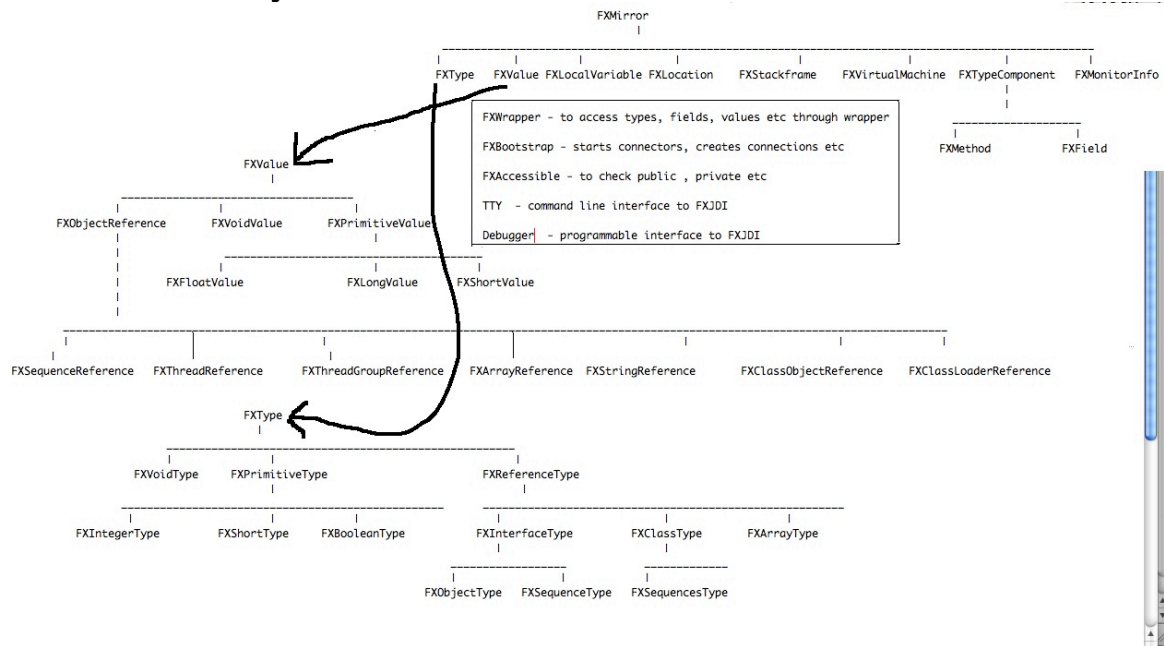
step , step(in/up), next, stop, trace, watch, where, list, dump, evaluate, threads etc .

1.1. Conventions

We follow a convention of including the target JavaFX code within the @BeginTest and @ EndTest and are single line commented , sample template is shown below

```
// @BeginTest Sample.fx
// function run():Void {
//     println("Test Ends here");
// }
// @EndTest
```

1.2. The class hierarchy/ structure of FXJDI



1.3. Scope

1.3.1 Whats in the scope

All the testcases must be woven around com.sun.javaafx.tools.debug.tty.Debugger which is programmatic interface to FX debugger

1.3.2 Whats out of scope

All the low level stuff falls out of our scope ,ones involving Java debugging wire protocol commands that shuttle between Target Virtual machine and debugger .

2. Tests inventory (Functional/Feature/Regression/Unit/etc.)

2.1. Existing tests

1) fxjdi.JdbBase - The base class for all the subsequent tests. This extends com.sun.javaafx.tools.debug.tty.Debugger and exposes some convenient methods like

- a) load, compile, compile0, fxrun the debuggee from a test directory
- b) methods to get hold, manipulate, compare, writeToFile the output of debug commands. For Ex: resetOutputs() redirects all the output to console. If this method is not called then by default all the out and err stream are used by ByteArrayOutputStream in JDbBase and not Console.
- 2) fxjdi.BasicTest - That hops through a set of 4 breakpoint and continues execution. At each breakpoint we validate by assertions.
- 3) fxjdi.LocalVarTest - Breakpoint set at point where local variable is not null, then asserted for not null.

2.2. New tests

New tests needs to be added to validate this feature as mentioned above. Please see section 4 for test case proposals.

2.3. Deprecate tests

- com.sun.java.fx.jdi.test contains all the test cases. All the test cases extends JavafxTestBase which extends the TestScaffold. JavafxTestBase has functionality to stop at breakpoint, step through, asserts at different events etc. All the outputs are directed to .ACTUAL and finally compares the .ACTUAL and .EXPECTED for PASS/FAILURE.

Helper classes

- TargetAdapter extends TargetListener - has callbacks to handle events like BreakPointEvent etc
- JDITestBase is used to basically attach connection(LaunchingConnector) to target VirtualMachine (debuggee).
- com.sun.java.fx.jdi.test.target contains all the test targets/debuggees.

3. Feature Cases, Tests and Test Cases

3.1. Feature Cases

Feature case ID	Feature case description	Eng. weeks estimation	Notes
CC_001			

3.2. Tests and Test cases

Test ID	Test case	Associated JIRA's ?	Description	# of testcases	# of manual and interactive
1	ScriptVarTest	JFXC-4412	Script level numeric variable, whose value is changed and checked at different points in the stack frame.	1	0
2	BasicBindTest		A script variable 'A' is bound to another variable 'B'. Changes to 'B' should propagate to 'A' and are asserted for validity.	1	0
3	WatchAccessModificationTest		watch all command is set on a static variable and is sent to target VM, so that we are notified about the access (AND) modifications occurring to the var.	1	0
4	HeteroObjectSequenceTest		Insert, delete objects of different types from a sequence, break at these points and assert what is anticipated.	1	0
5	SequenceBindTest		bind on a sequence	1	0
6	ComplexBrkpointTest		Set break points at multiple stack frames	1	0
7	TraceCallsTest		Trace the entry and exit of the methods in the FX script	1	0
8	StepInUpTest		Simulates a "step in" to another stack frame and "step out" to main stack frame	1	0
9	StopAtBkpointTest	JFXC-4419	A particular use case which was skipping past a breakpoint. NOTE: All the waitForBreakpoint() calls have been replaced with resumeToBreakpoint()	1	0
10	BoundTest	JFXC-4455	Break at specific points in script with bound function, the LNT entries seem to be mangled and the breakpoints on line numbers are not set correctly.	1	0

11	BreakAtOnReplaceTest		Set a breakpoint inside the onreplace trigger. Whenever the binder value is changed ensure that the executions halts inside the trigger and resumes again.	1	0
12	FrameFilterTest			1	0
13	DumpTest		Simple Test to dump object values . Added convenient method containsInTandem() to JdbBase for this purpose.	1	0
14	ExprEvaluateTest		Evaluate an expression through Debug API for FXJDI . For instance try a recursive function call in expression and evaluate it.	1	0
15	EventManagerTest		A scenario that tests all the event requests managed are going through EventRequestManager .	1	0
16	BasicInheritanceTest		Class Child inherits Sooper,	1	0
17	MixinTest				0
18	MultiThreadBreakTest		CreateMultiple FX Thread tasks , break at certain point in each thread and assert for ThreadReference etc	1	0
19	JumpLocationsTest		Have multiple stack frames , jumping from one to another and unwinding all the calls in the last. Make sure the debugger stops at expected points and verify the location and its attributes.	1	0
20	MethodUnwindTest		Make a series of nested method calls and pop off the most recent stackframe from the call stack and validate you are on the right stackframe.	1	0
21	ReentrantTest		This use case can be seen as an extension of MethodUnwindTest , where in now you also step upto the last stack frame again .	1	0
22	ExceptionCatchTest		Try to throw an exception without a try catch block. Tell the debugger to catch the Exception which anyways breaks after the exception event is caught, then we analyze it.	1	0

3.3. User Scenarios (Optional)

N/A

4.0. Test Implementation Notes

4.1. Test development notes (Optional)

- All the test methods are annotated with `@Test(timeout=5000)` **NOTE** : The timeout can be changed.
- If the test method starts with `resetOutputs()` then all the console output and error stream goes to console. This can be used for debugging your testcase. We should comment this method when we push the TC to repo.
- When a JIRA bug is raised for a scenario, and we have a TC for it, then we must comment the annotation `@Test` to make sure the Test is not run by the build system else it fails. The engineer after fixing the bug must take the responsibility of un-commenting the annotation.
- All the review requests are webrevs and are available at <http://kal.sfbay.sun.com:8080/rb/fxjdi/> . For webrev templates please goto section 6.6 .
- Added the util methods to JdbBase

`verifyLiteralValue(String var, String expectedValue)` - compares (var == expectedValue)

`verifyNumValue(String var, Number expectedValue)` - compares (var == expectedValue)

`containsInTandem(String[] expected)` - This method checks if the given set of strings appear in the output, in the same given order . If the order is broken then it is deemed not contained. So this method is not similar to `Collection.containsAll` which doesn't care for order.

4.2. Code dependencies

- All the test cases in the package `fxjdi` (Tests folder) are Junit based and hence make sure junit is in the classpath.
- The `fxjdi` project can be imported to as netbeans project as it has necessary project files . Then you can just right click on each individual tests and run the test .

4.3. Build and setup procedures

- First build the openjfx-compiler~soma-patch , make sure JAVA_HOME points to 1.6 (1.5 will not work!!) . Do **ant clean** and then **ant** .
- If you have only made changes to fxjdi sub project then you can build it exclusively as
ant fxjdi (OR) Go into the fxjdi subfolder , run **ant**

4.4. Manual and interactive test justification

TBD

4.5. Assumptions and dependencies

TBD

4.6. Tips for Debugging

- 1) javap classname - Gives a list of variables and method names
- 2) javap -v classname - Gives a list of all fields, methods, line number table(lineNO -> ByteCode mapping)
- 3) javafx -XDdumpjava Sample.fx - Dumps the intermediate java code in a folder (typically dumpjava)
- 4) run(), resume - are all asynchronous calls and hence you need to call waitForBreakpointEvent() to hang and wait till you receive Event(Not required for command line debugging) .

4.7. Associated Bugs

JIRA Bug Id	Description	Raised Date	Resolution Date	Status
JFXC-4412	next() does not hit the last line and stop	6/9/2010	NA	In Progress
JFXC-4419	Debugger skips past a breakpoint	6/10/2010	6/14/2010	Resolved
JFXC-4455	LNT entries are missing when using bound function	6/25/2010	NA	New

5.0. Test Configurations

5.1. Operating Systems:

TBD

#	OS
01	Windows XP, Vista, 7
02	Mac OSX Leopard, snow leopard

6. Resource Section

6.1. Hudson job:

<http://openjfx.java.sun.com/hudson/job/openjfx-compiler/>

Builds just the compiler from our kenai repo: ssh://<username>@hg.kenai.com/openjfx-compiler~soma-patch whenever a push to the compiler occurs.

6.2. Compiler scrum hudson job:

<http://jfx.sfbay.sun.com/hudson/job/1.3.1-compiler-scrum/>

ssh://<username>@jfxsrc.sfbay.sun.com/javafx/1.3.1/scrums/compiler/jfx/ . This repo normally just gets modified when an integrated into MASTER.

6.3. Kumar's hudson job:

Clones the compiler scrum forest(in hudson), pull in the current compiler bits from kenai, and build the result.

6.4. Latest compiler builds + sdk:

1) (Windows): <http://wallowa.sfbay.sun.com:8081/job/openjfx-forest/lastSuccessfulBuild/artifact/artifacts/bundles/>

2) (Mac): <http://pasilla.sfbay.sun.com:8081/job/openjfx-forest/lastSuccessfulBuild/artifact/>

3) fxjdi: If you download the SDK , the fxjdi jar is in lib/shared folder. Else you can get it from <http://wallowa.sfbay.sun.com:8081/job/openjfx-compiler/lastSuccessfulBuild/artifact/fxjdi/dist/fxjdi.jar>

6.5. NetBeans:

<http://big-mac.czech.sun.com/hudson/job/JavaFX%20Composer%20FXJDI>

This means that if you have downloaded and been running NB and you want to switch to a newer fxjdi.jar, you build fxjdi.jar yourself or whatever, and cp it into nb/javafx/modules/ext/fxjdi.jar . When you cp fxjdi.jar into NB, you could ALSO cp it into the javafx SDK inside NB. Then you could use that SDK to run normal javafx/ jdb stuff. Or you could do what I do and just build your own forest:

```
hg fclone ssh://<username>@jfxsrc.sfbay.sun.com/javafx/1.3.1/scrums/compiler/jfx/ kkk
```

```
cd kkk/openjfx-sdk
```

```
hg pull ssh://<username>@hg.kenai.com/openjfx-compiler-soma-patch
```

```
cd ..
```

```
ant to build it all
```

Then you still have to remember to cp the built fxjdi.jar into that special NB area.

6.6. Web Rev template

- Download it [here](#) .

7. Responsibilities/Contacts

Action/Role	Contact Info
Draft Prepare/Test Developer/Test Lead	Srikalyan Chandrashekar
JavaFX compiler Lead	Robert Feild
SQE Manager	Ken Chen / Sandeep Konchady

8. Approvals

9. Revision History

Version	Author(s)	Comments	Date
1.0	Srikalyan Chandrashekar	Initial draft	06/01/2010
2.0		Moved the doc from compass.sfbay to sunspace wiki .	06/10/2010
3.0		More Test implementation notes	06/14/2010