



Building a Java™ ME Test Suite in 15 Minutes

Mikhail Gorshenev, Senior Staff Engineer
Roman Zelov, Member of Technical Staff
Alexander Glasman, Member of Technical Staff
Sun Microsystems, Inc.
<http://www.sun.com/>

TS-5906

Goals

Understand the complexity of mobile application testing.

Learn how to build test suites for small devices.

Agenda

Mobile Application Testing Challenges

Best Practices in Testing

Testing Tools and Initiatives

Organizing and Managing Tests

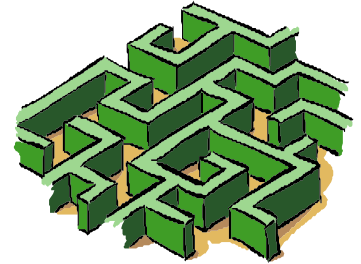
Writing Advanced Tests

Demo

Testing on Small Devices

Why It Is Different

- Device constraints
 - Small screen
 - Limited memory
 - Few threads
 - Intermittent connectivity
 - No/restricted file system
- Test automation
 - Hard to run multiple tests over slow connections
 - MIDlet instantiation: can't call `new MIDlet()`
- On-device debugging
 - No standardized debugging interface



Mobile Application Testing



Scalability Issues

- Application porting difficulties: fragmentation
 - Growing number of mobile devices with different characteristics
 - Many technologies (Java Specification Requests (JSRs))
 - Multiple platforms (CLDC, CDC) and versions
 - Optional features, OEM-specific classes
 - Implementation bugs
 - Large testing matrix: Devices x Applications
- Managing the testing process
 - Unit tests good for standalone applications
 - More powerful solution required for the enterprise



Agenda

Mobile Application Testing Challenges

Best Practices in Testing

Testing Tools and Initiatives

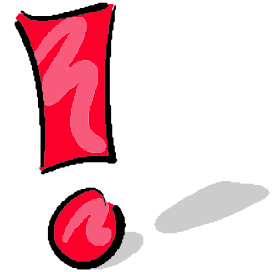
Organizing and Managing Tests

Writing Advanced Tests

Demo

Testing Best Practices

- Start early
- Write unit tests
 - Avoid duplication: one test is better than 20
- Design for testing
 - Make your process repeatable
 - Automate!
- Measure quality
 - Track your progress
 - Set up reporting
 - Use bug tracking tools



Choose the Right Test Environment

Device vs. the emulator

- Device testing
 - Slow connection
 - Limited automation
 - Operator-controlled network environment
- Emulator
 - Faster testing, saves time
 - May not reflect real device behavior
 - Address book, phone calls
 - Bluetooth, Location APIs



Agenda

Mobile Application Testing Challenges

Best Practices in Testing

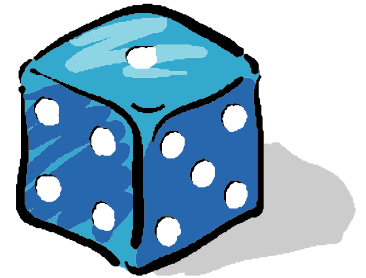
Testing Tools and Initiatives

Organizing and Managing Tests

Writing Advanced Tests

Demo

Testing Tools



- NetBeans™ Mobility Pack for CLDC
 - IDE for development and debugging
 - JUnit, J2MEUnit test framework
- Sun Java Wireless Toolkit
 - Debugger
 - Profiler
 - Network monitor
 - Memory monitor
- java.net Mobile&Embedded community tools
 - JT Harness
 - ME Framework

Sun Java Wireless Toolkit Tools

Debugging tools

The screenshot displays three debugging tools from the Sun Java Wireless Toolkit:

- Methods Profiler:** Shows a call graph and a table of all calls under the root. The table lists method names, counts, and cycles.
- Network Monitor:** Displays a list of HTTP requests and responses, including status codes and headers. A detailed view of a request is shown on the right.
- Memory Monitor Extension:** Shows a table of memory usage for various objects and a graph of memory usage over time.

Name	Count	Cycles
<root>	0	0
com.sun.midp.lcdui.DefaultEventHandler\$...	0	70719316
example.payment.jbricks.Screen.paint	14193	34475365
example.payment.jbricks.Screen.paintMenu	12513	50571769
javax.microedition.lcdui.Graphics.drawStri...	187673	12836074
example.payment.jbricks.Ball.paint	1462	11113914
javax.microedition.lcdui.Graphics.drawArc...	2924	59265857
com.sun.midp.lcdui.DefaultEventHandler.r...	14188	10623131
example.payment.jbricks.Brick.paint	1869	16294504
javax.microedition.lcdui.Graphics.drawLine...	14576	488835158
example.payment.jbricks.Menu.getItems	473749	229485403
example.payment.jbricks.Engine.run	0	48335840
example.payment.jbricks.Screen.paintTitle	212	1687312
com.sun.midp.midlet.Selector.run...	0	25570
example.payment.jbricks.Main.<init>	0	22375008
java.lan.Math.min...	88	116882895

Method	Status
POST /servicescript/servicescript HTTP/1.1	200 OK
HTTP/1.1	200 OK
POST /servicescript/servicescript HTTP/1.1	200 OK
HTTP/1.1	200 OK
POST /servicescript/servicescript HTTP/1.1	200 OK
HTTP/1.1	200 OK
POST /servicescript/servicescript HTTP/1.1	200 OK
HTTP/1.1	200 OK

Name	Live	Total	Total Size	Averag...
VM Internal	206	1942	15104	73
java.lang.OutOfMemoryError	1	1	20	20
java.lang.String[]	77	131	3128	40
java.lang.Thread	5	5	140	28
char[]	603	18026	59044	97
java.io.PrintStream	1	1	28	28
com.sun.midp.io.SystemOutp...	1	1	12	12
java.io.OutputStreamWriter	1	1	28	28
java.lang.String	905	2086	21720	24
java.lang.StringBuffer	307	505	7368	24
com.sun.cldc.i18n.ucl.Default...	1	1	12	12
java.lang.ClassNotFoundExc...	0	2	0	0

JMUnit, J2MEUnit

Unit testing framework

- Open source unit testing frameworks
- Based on JUnit tool for Java Platform, Standard Edition (Java SE platform)
- Support CLDC 1.0/1.1, MIDP 1.0/2.0
- Not compatible with each other
 - Plans to merge the future versions into one
- Quick to learn, easy to use
- Limited automation and test management
- Can't handle large test suites

JT Harness and ME Framework

Open source testing harness

- cqME™ project (<http://cqme.dev.java.net>)
- JT Harness
 - Based on Sun JavaTest™ Harness, one of the first Java applications in the world
 - In use for conformance testing since 1995: all Java platform implementations certified used this harness
 - Suited for most types of tests (unit testing and more)
 - Supports Java SE platform and Java Platform, Enterprise Edition (Java EE platform)
- ME Framework
 - Java ME platform plug-in for JT Harness
 - Based on Sun's Java ME TCK framework
 - Supports all versions of CLDC, MIDP, CDC/FP, PBP/PP

Unit Test Example (JMockUnit)

```
public void testadd() throws AssertionFailedException {  
    System.out.println("add");  
    math.Arithmetic instance = Arithmetic.getInstance();  
    for (int a=0;a<=10;a++)  
        for (int b=0;b<=10;b++) {  
            int expectedResult = a+b;  
            int result = instance.add(a,b);  
            assertEquals(expectedResult, result);  
        }  
}
```

Unit Test Example (JT Harness)

```
public Status testadd() {
    log.println("add");
    math.Arithmetic instance = Arithmetic.getInstance();
    for (int a=0;a<=10;a++)
        for (int b=0;b<=10;b++) {
            int expectedResult = a+b;
            int result = instance.add(a,b);
            if (expectedResult != result)
                return Status.failed("Invalid result");
        }
    return Status.passed("OK");
}
```

Industry Initiatives

Improving quality, reducing fragmentation



- **Sun Java Device Test Suite**
 - Defragmentation test suite
- **Quality guidelines specifications from vendors and standards bodies; Example: 3G Americas**
 - Platform test suite is developed in open source (<http://cqme.dev.java.net>)
- **Java Verified™ Program**
 - Application testing and digital signing program for Java ME platform applications
 - Trusted third-party signature
 - Promotion through vendor/operator catalogs

Agenda

Mobile Application Testing Challenges

Best Practices in Testing

Testing Tools and Initiatives

Organizing and Managing Tests

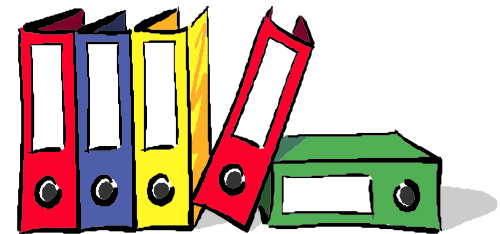
Writing Advanced Tests

Demo

Running Large Number of Tests

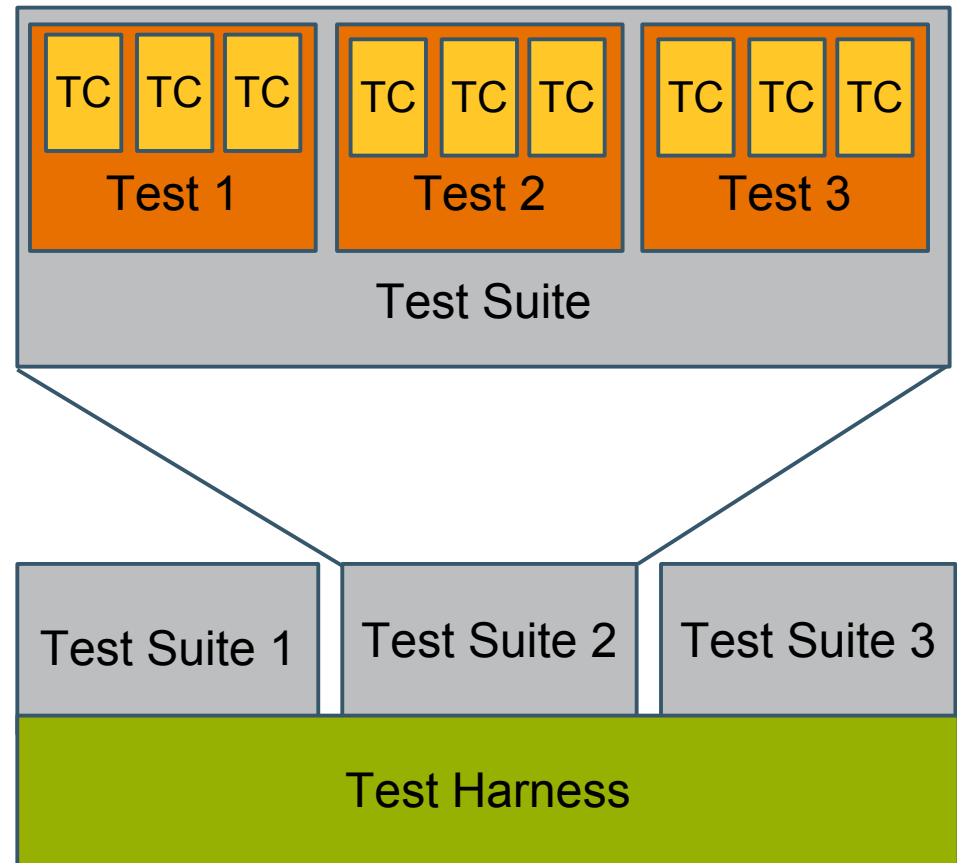
Typical difficulties

- Deploying tests
 - Device constraints
 - Scalability
 - Reliability
 - Test interference
- Managing tests and results
 - Configuration
 - Logging and debugging
 - Reporting
- Shortening the test cycle



Organizing Tests

- Test cases
- Tests
- Test suites
- Test harness



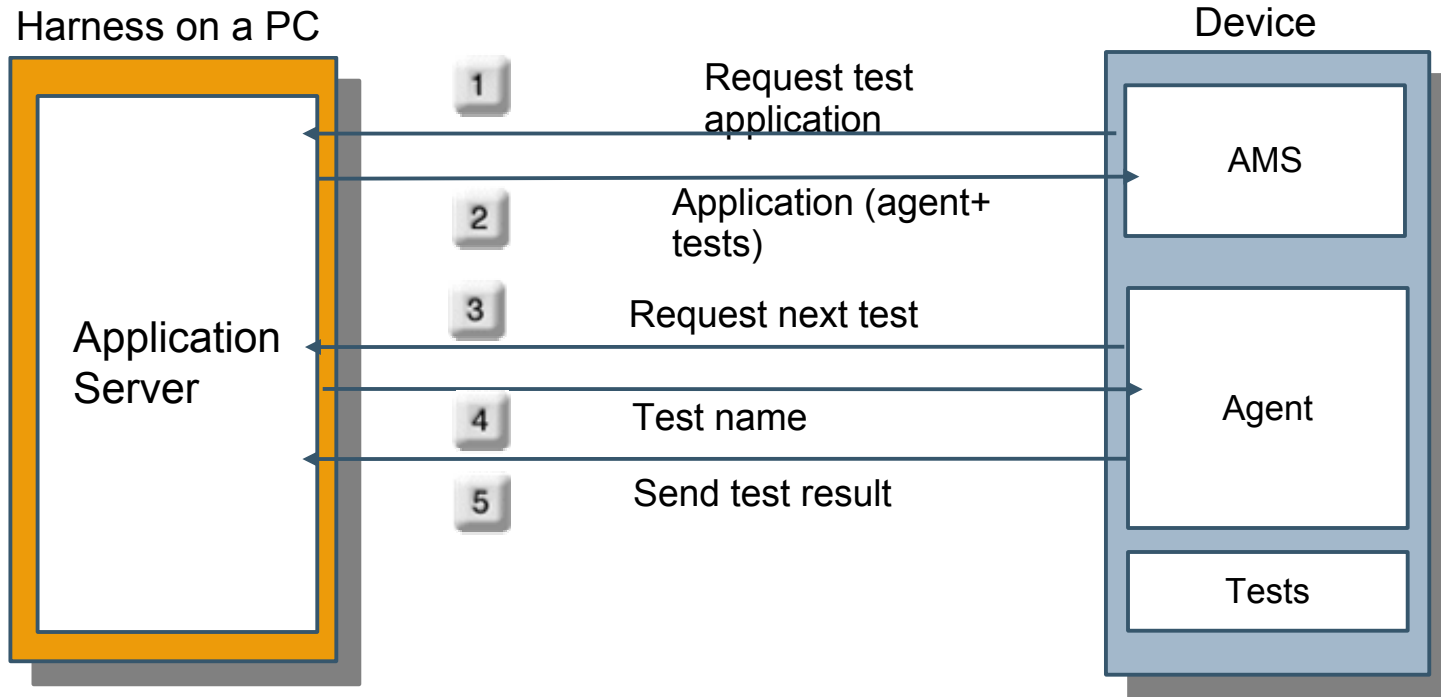
Solutions for Device Testing

- Harness-agent architecture
- Automation: autotest (limited support)
- Application provisioning support
 - Application server
 - Packaging: Java Archive (JAR) file, Java Application Descriptor (JAD) file, manifest
 - Application signing
 - Security
- Optimization
 - Test bundling
 - Agent pre-installation
 - Parallel testing



Test Automation

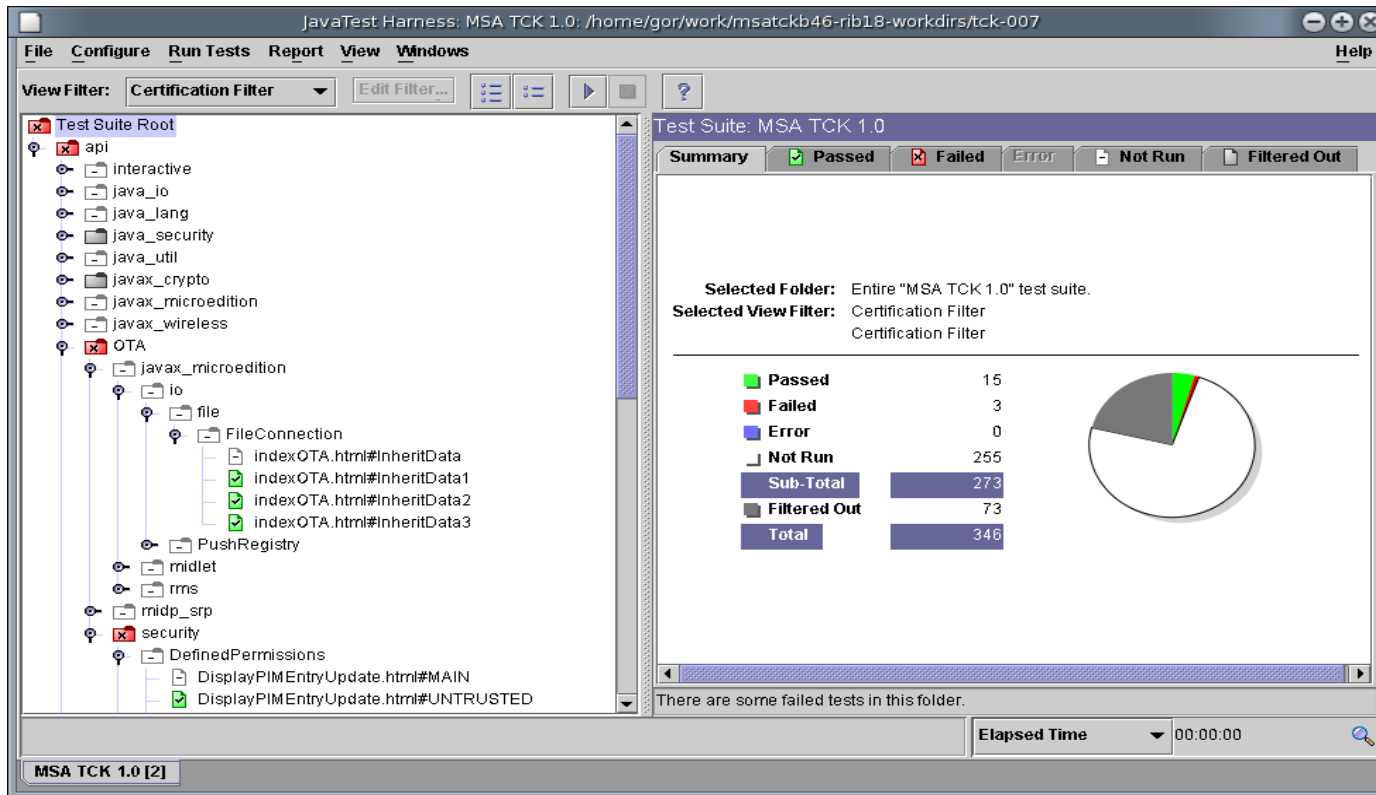
Harness-agent approach



Managing Tests

JT Harness Test Manager GUI

- Designed for the end user (tester)



The screenshot shows the JavaTest Harness GUI for MSA TCK 1.0. The interface includes a menu bar (File, Configure, Run Tests, Report, View, Windows, Help), a toolbar, and a 'View Filter' dropdown set to 'Certification Filter'. On the left is a tree view of the test suite structure, with 'OTA' selected. The right pane displays the 'Test Suite: MSA TCK 1.0' summary, including a table of results and a pie chart.

Category	Count
Passed	15
Failed	3
Error	0
Not Run	255
Sub-Total	273
Filtered Out	73
Total	346

Selected Folder: Entire "MSA TCK 1.0" test suite.
 Selected View Filter: Certification Filter

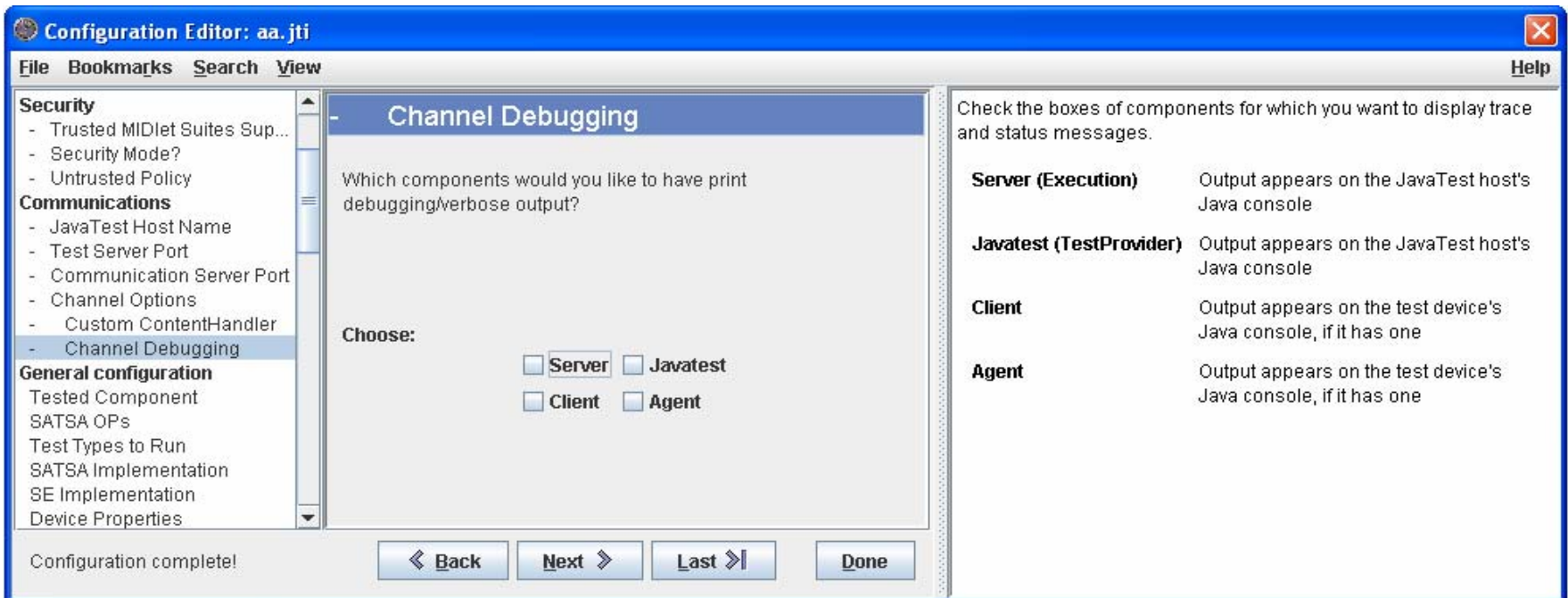
There are some failed tests in this folder.

Elapsed Time: 00:00:00

Configuration

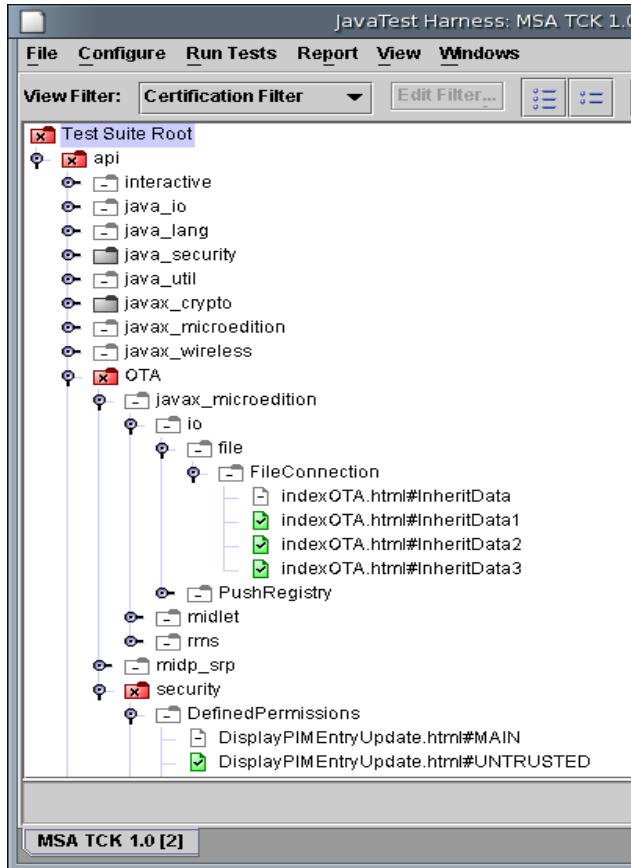
JT Harness Configuration Editor

- Simple configuration wizard for the users
- Flexible API for developers



Reporting

Report Browser



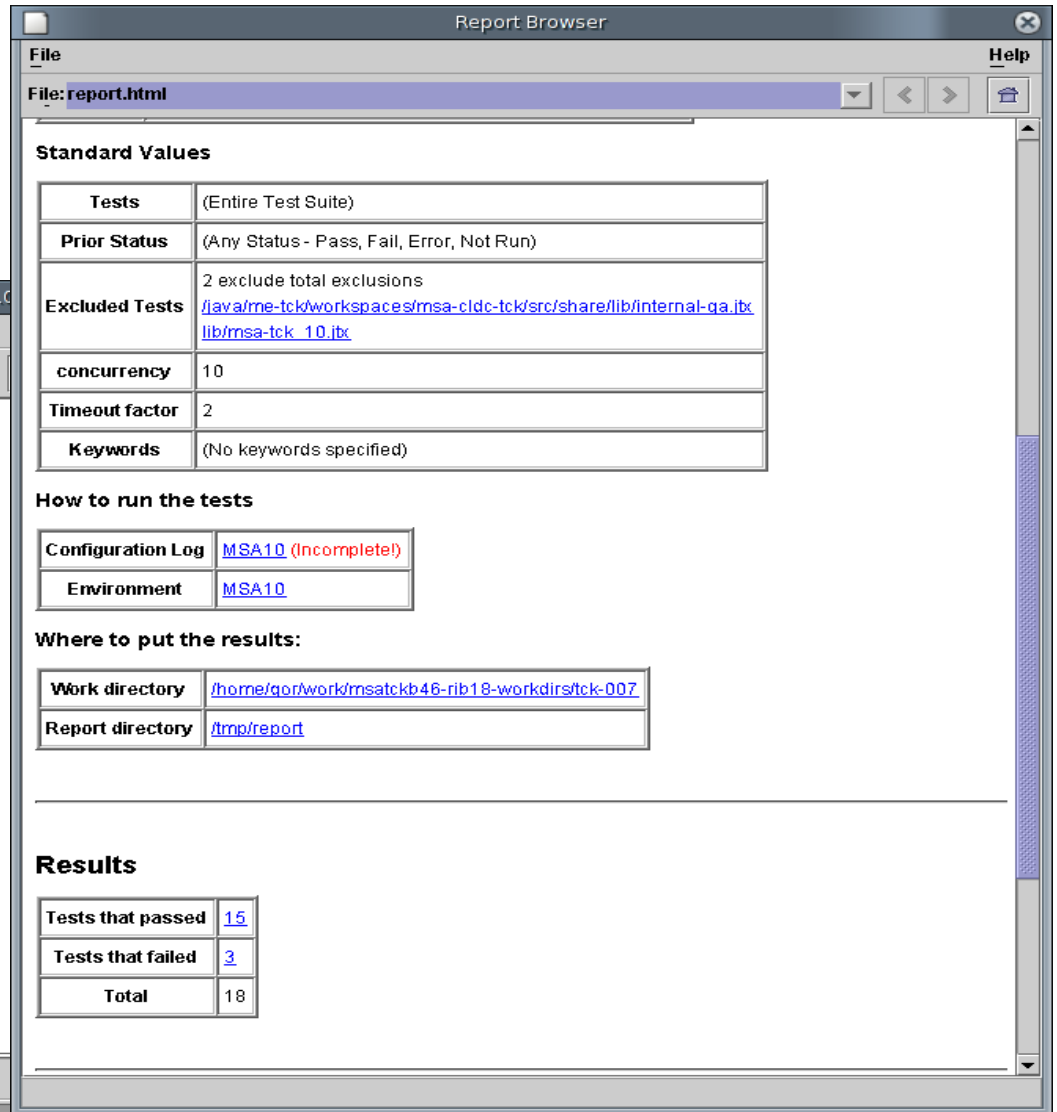
JavaTest Harness: MSA TCK 1.0

File Configure Run Tests Report View Windows

View Filter: Certification Filter Edit Filter...

- Test Suite Root
 - api
 - interactive
 - java_io
 - java_lang
 - java_security
 - java_util
 - javax_crypto
 - javax_microedition
 - OTAs
 - javax_microedition
 - io
 - file
 - FileConnection
 - indexOTA.html#InheritData
 - indexOTA.html#InheritData1
 - indexOTA.html#InheritData2
 - indexOTA.html#InheritData3
 - PushRegistry
 - midlet
 - rms
 - midp_srp
 - security
 - DefinedPermissions
 - DisplayPIMEntryUpdate.html#MAIN
 - DisplayPIMEntryUpdate.html#UNTRUSTED
 - javax_wireless

MSA TCK 1.0 [2]



Report Browser

File Help

File: report.html

Standard Values

Tests	(Entire Test Suite)
Prior Status	(Any Status - Pass, Fail, Error, Not Run)
Excluded Tests	2 exclude total exclusions /java/me-tck/workspaces/msa-cldc-tck/src/share/lib/internal-ga.lib/lib/msa-tck_10.lib
concurrency	10
Timeout factor	2
Keywords	(No keywords specified)

How to run the tests

Configuration Log	MSA10 (Incomplete)
Environment	MSA10

Where to put the results:

Work directory	/home/gor/work/msatckb46-rib18-workdirs/tck-007
Report directory	/tmp/report

Results

Tests that passed	15
Tests that failed	3
Total	18

Agenda

Mobile Application Testing Challenges

Best Practices in Testing

Testing Tools and Initiatives

Organizing and Managing Tests

Writing Advanced Tests

Demo

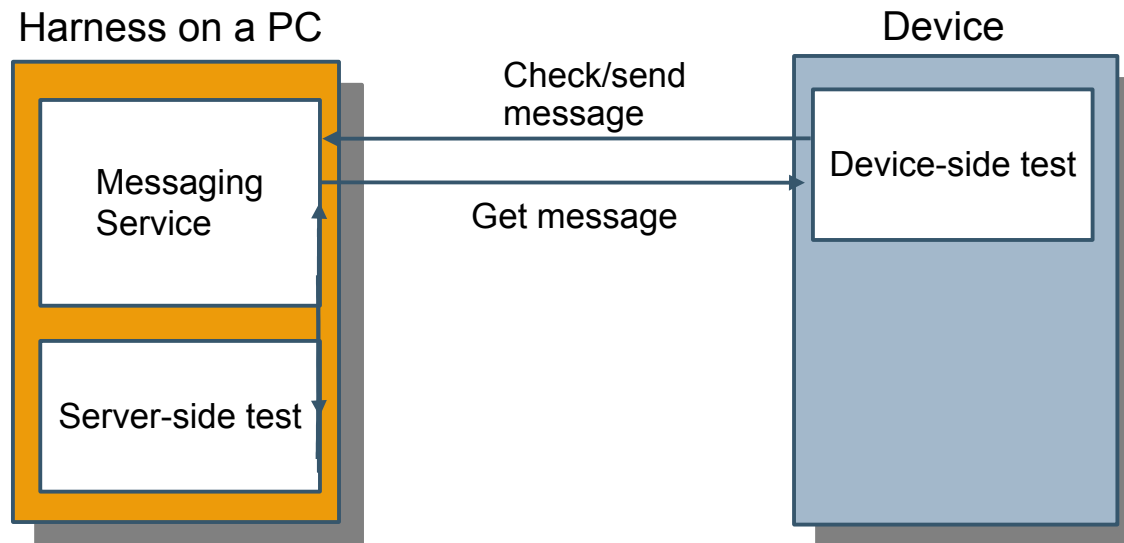
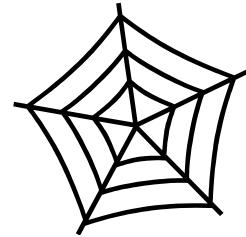
Advanced Testing

ME Framework Extensible Test Libraries

- Support for testing needs beyond basic unit tests
- Distributed tests
 - Verify networking, communication functionality
- Interactive tests
 - When automated testing is not enough (user interface)
 - Use the “big screen” for instructions when possible
- Security tests
 - Validate security-constrained operations
- OTA tests
 - Over-the-air provisioning of applications

Distributed Test

- Two (or more) tests run on different hosts
 - Two Java ME platform devices
 - Java ME platform device and Java SE platform harness machine
- Messaging API for synchronization



Interactive Test

OpenGL ES Translate Test

Test Instructions

This test verifies the translation matrix operations in OpenGL ES implementation.
Click on each Test button and make sure that the tested implementation displays the image similar to the corresponding snapshot taken from the reference implementation.
Note: The test verifies the GL10.glTranslatef(float x, float y, float z) method.

Test 1: The cube is in the center of the screen. Translation matrix is (0, 0, -60*edge).

Test 2: The cube is on the right side, partially cropped by the right clipping plane. Translation matrix is (2*edge, 0, -60*edge).

Test 3: The cube is on the top of the screen, partially cropped by the top clipping plane. Translation matrix is (0, 2*edge, -60*edge).

Test 4: The cube is moved deep into the screen. Translation matrix is (0, 0, -260*edge).

Timer: Time remaining: 02:58 [Pause]

Does it appear like on the snapshots?
[Yes] [No]

+5550000 - DefaultColorPhone
IDlet Help



DEMO

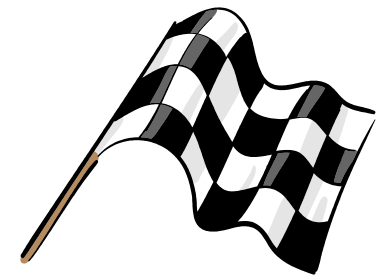
Building a Java ME Test Suite in 15 Minutes

Summary

- Mobile application testing is a complex endeavor
- Small devices require large test suites
- Powerful tools help with the task

For More Information

- <http://www.netbeans.org/products/mobility/>
- <https://jtharness.dev.java.net/>
- <https://cqme.dev.java.net/framework.html>
- <http://javaverified.com>





Q&A





Building a Java™ ME Test Suite in 15 Minutes

Mikhail Gorshenev, Senior Staff Engineer
Roman Zelov, Member of Technical Staff
Alexander Glasman, Member of Technical Staff
Sun Microsystems, Inc.
<http://www.sun.com/>

TS-5906