

Paradise lost: Troubleshooting Java™ applications outside of development sandbox

Alexandre Rafalovitch

May 2006

Learn how to analyze and resolve
production Java problems
without panic

For the next 60 minutes

Why are you here? Why am I here?

Descriptive, not prescriptive approach

System boundaries and choke points

Tools and analysis methods

Proficiency across platforms

Honing the troubleshooting skills

Looking forward

Let's talk about

Why are you here? Why am I here?

Descriptive, not prescriptive approach

System boundaries and choke points

Tools and analysis methods

Proficiency across platforms

Honing the troubleshooting skills

Looking forward

Why are you here? Why am I here?

- You
 - Technical support trying to upgrade skills
 - Programmer looking for better troubleshooting tools
 - Manager desiring to improve *technical* support
- Me
 - 3 years as BEA senior tech. support engineer (DRE)
 - Java professional since JDK1.0b2
 - Not a guru – just sharing the collected experience

Let's talk about

Why are you here? Why am I here?

Descriptive, not prescriptive approach

System boundaries and choke points

Tools and analysis methods

Proficiency across platforms

Honing the troubleshooting skills

Looking forward

Descriptive, not prescriptive approach

- Prescriptive approach is wishing for a silver bullet
 - One cannot ask developer to 'never do something'
 - Most complex problems are emergent issues
 - Expensive software is great, but is usually too late
- Descriptive approach is about understanding
 - Somebody will always end up doing X
 - And sometimes it is just '*Dude, Where is my log file?*'
 - Even if you don't know how you got there, you still have to fix it

Descriptive, not prescriptive approach

Problem with programmer's method

- *System.out.println()* is NOT your friend in production
- Recompile and restart is infrequently an option
- Autowiring is good until something gets miswired
- Did anyone tell you about the firewall?
- When the system is losing more per hour than you make per year – this is not the time to start reading APIs
- Things are getting better, but production still runs old stuff

Let's talk about

Why are you here? Why am I here?

Descriptive, not prescriptive approach

System boundaries and choke points

Tools and analysis methods

Proficiency across platforms

Honing the troubleshooting skills

Looking forward

System boundaries and choke points

or knowing where to look

- Modern programs are beyond *cat*, *grep* or *sort*
 - Require configuration files
 - Create logs
 - Run in grids and clusters
- *Filesystem boundary* (config, log, classpath)
- *Network boundary* (clusters, webapps, JDBC)
- *Processor and memory* (multithreading)
- *Environment variables* (OS/user specific)
- *Configuration files* (XML, properties, automagic)

Example program

```
public static void main(String[] args) throws Exception {
    Preferences prefs =
        Preferences.userRoot().node("Boundaries");
    int port = prefs.getInt("port", 8001);
    int idx=0;
    ServerSocket socketListener = new ServerSocket(port);
    Logger logger = Logger.getLogger("Boundaries");
    while(true) {
        Socket socket = socketListener.accept();
        logger.info("Accepted connection: " + idx);
        BufferedReader in = new BufferedReader(
            new InputStreamReader(socket.getInputStream()));
        FileWriter writer = new FileWriter(args[idx++]);
        String line;
        while ((line = in.readLine()) != null) {
            writer.write(line); writer.write('\n');
        }
    }
}
//missing something?
```

System boundaries and choke points

Filesystem

- Ignore relative path puzzle, look at lower level
- Currently open files (logs, locks, jars)
 - More than you expect
 - Process Explorer on Windows, lsof on *nix
- Transient files and file search (configs, classpath)
 - The system is way busier than you expect
 - Performance lessons of classpath ordering
 - FileMon on Windows, trace/struss/dtrace on *nix

System boundaries and choke points

Network

- Applications are becoming more and more *chatty*
 - Webapps with a browser as a platform
 - Webstart applications
 - AJAX (quantitative change)
 - Clustering
 - JDBC
- Most of the traffic is over HTTP
- A lot of troubleshooting information is available, but it is hard to see with all the layers on the stack
 - Ethernet, IP, TCP, HTTP, XML

System boundaries and choke points

Processor and memory

- Processes are becoming more instrumentable
- Still hard to look inside, but getting better
- JVMs expose more information via JMX
- Multithreading issues will become more prominent
- Java thread dumps
 - There are problems with JIT and different JVMs
 - Locking information was good in 1.4, incomplete in 5.0, improved in 6.0

System boundaries and choke points

Environment variables

- If something is not defined anywhere in the program, look in the environment
 - JVM version
 - Default classpath
 - Extensions jars
- Different in Windows and Unix
 - Everything is in the files on Unix
 - Windows can have it in files or in registry

System boundaries and choke points

Configuration files

- GUI configuration does not survive meeting the troubleshooting reality
 - Compare settings across servers
 - Ultimate authority, when something is wrong
- Usually there more files than expected
 - Tomcat has 23 XML files (9 types) + 3 .properties
- Good news: configuration files are parsable
 - Parsable means they can be correlated
- Some things are not in the configuration files
 - Autowiring
 - Defaults

Let's talk about

Why are you here? Why am I here?

Descriptive, not prescriptive approach

System boundaries and choke points

Tools and analysis methods

Proficiency across platforms

Honing the troubleshooting skills

Looking forward

Tools and analysis methods

Modern JVM and its advantages

- JDK 5.0 and JDK 6.0 new troubleshooting tools
 - jconsole, jps, jhat, jmap, jstack, jstat
- If your JVM does not cut it, look at others
 - BEA JRockit – memory leak detector, console
- Look at what your O/S comes with
 - DTrace on Solaris
- 3rd party tools
 - Easy install – too late for complex configurations
 - Minimum admin privileges – not always possible

Tools and analysis methods

Filesystem boundary

- Currently open
 - Log files, active IO, leaking handles
 - Windows: ProcessExplorer/Handle from Sysinternals
 - *nix: lsof
- Solves
 - Resolves relative paths
 - Shows leaking handles
 - Default locations for log files
- From our example:

```
FileWriter writer = new FileWriter(args[idx++]);
```

[-] [eclipse icon] eclipse.exe	1...	
[-] [javaw icon] javaw.exe	896	Java(TM) 2 Platform Star
[-] [javaw icon] javaw.exe	1...	Java(TM) 2 Platform Star
[cmd icon] CMD.EXE	1	Windows Command Prom

T...	Name
File	C:\Projects\Eclipse\JavaOneExamples
File	\Device\NamedPipe\Win32Pipes.00000...
File	C:\TEMP\result2.txt
File	\Device\Tcp
File	\Device\Tcp
File	\Device\Afd\Endpoint
File	C:\TEMP\result.txt
...	...

Tools and analysis methods

Filesystem boundary

- Files briefly accessed by the process
 - Configuration files, classpath checking, jsp reload
 - Windows: FileMon from Sysinternals
 - *nix: truss/strace
 - Solaris 10: dtrace
- Solves
 - Configuration files not where expected
 - Incorrect library version is picked up
 - File (JSP) changed but not reloaded
- From our example:
 - Let's look at classpath

Tools and analysis methods

- FileMon output showing classpath search

```
javaw.exe... QUER... C:\TEMP\extraclasspath SUCCESS
javaw.exe... QUER... C:\Program Files\Java\jre1.5.0_06\lib\ext\dnsns.jar SUCCESS
javaw.exe... QUER... C:\Program Files\Java\jre1.5.0_06\lib\ext\localedata.jar SUCCESS
javaw.exe... QUER... C:\Program Files\Java\jre1.5.0_06\lib\ext\sunjce_provider.jar SUCCESS
javaw.exe... QUER... C:\Program Files\Java\jre1.5.0_06\lib\ext\sunpkcs11.jar SUCCESS
javaw.exe... QUER... C:\Projects\Eclipse\JavaOneExamples\example\Boundaries.class SUCCESS
```

Tools and analysis methods

Network boundary

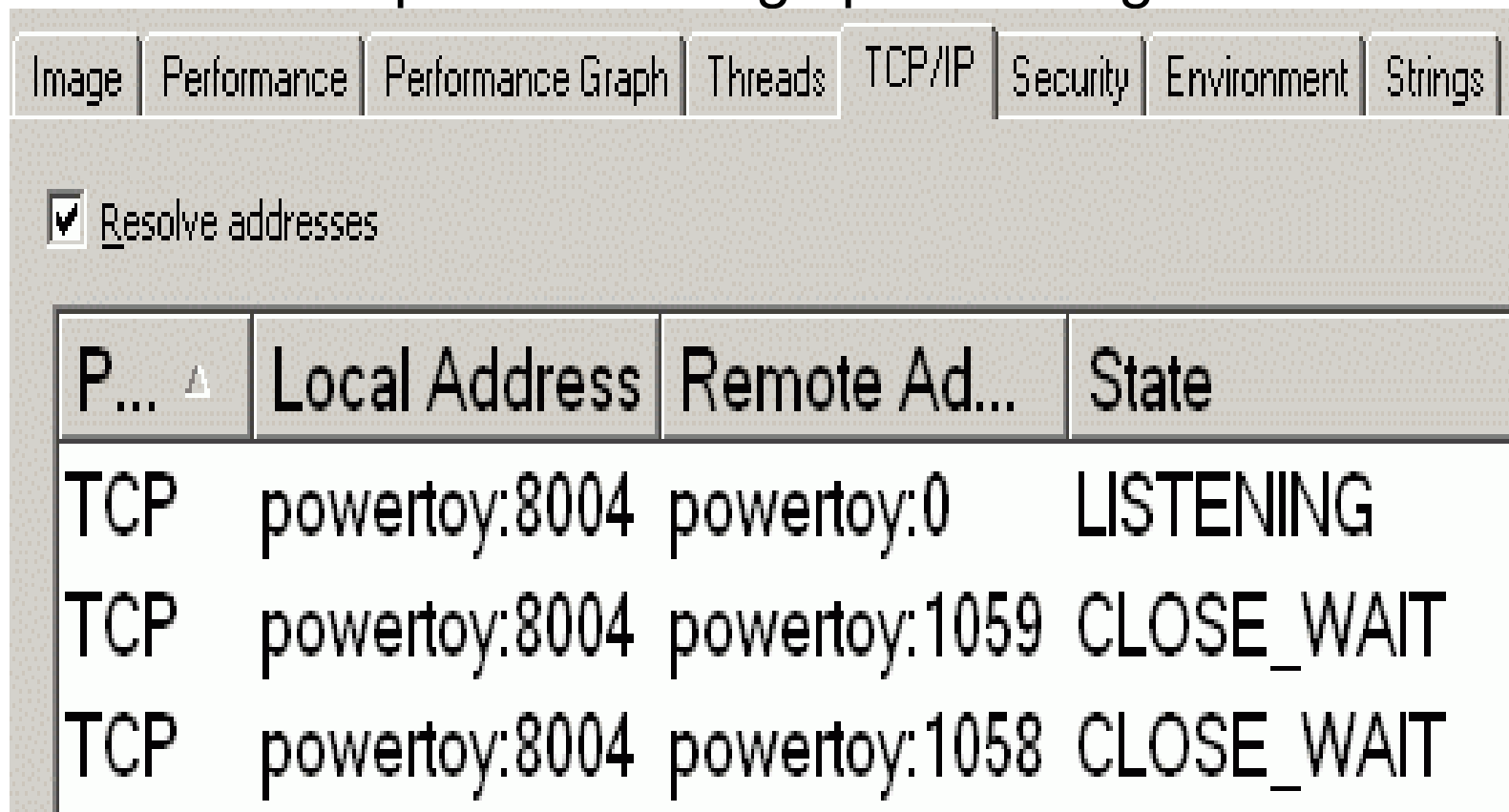
- Currently open connections
 - Unknown configuration, leaking descriptors
 - Same approach as with currently open files
 - Windows: ProcessExplorer/TCPView from Sysinternals
 - *nix: lsof
- From our example:

```
ServerSocket socketListener = new ServerSocket(port)....
```

```
Socket socket = socketListener.accept();
```

Tools and analysis methods

- ProcessExplorer showing open/leaking sockets



The screenshot shows the TCP/IP tab in Process Explorer. The 'Resolve addresses' checkbox is checked. The table below displays the open sockets for the 'powertoy' process.

P... ▲	Local Address	Remote Ad...	State
TCP	powertoy:8004	powertoy:0	LISTENING
TCP	powertoy:8004	powertoy:1059	CLOSE_WAIT
TCP	powertoy:8004	powertoy:1058	CLOSE_WAIT

Tools and analysis methods

Network boundary

- Network traffic over time
 - webapps, applets, Cluster replication, JDBC, LDAP
 - Windows/*nix/*: Ethereal
 - Open source and multi-platform
 - Reads >20 tracer/tcpdump formats
 - Parses > 750 protocols (including HTTP and XML)
 - Custom capture/display filters
 - Displays both high and low level details as needed
 - Can be installed on client,server or spanning port
- A high/low level example
 - Connect to <http://www.news.com>

Tools and analysis methods

```
⊞ Frame 7 (490 bytes on wire, 490 bytes captured)
⊞ Ethernet II, Src: Netgear_64:ce:6e (00:0f:b5:64:ce:6e)
⊞ Internet Protocol, Src: 216.239.115.148 (216.239.115.148)
⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: http (80)
⊞ Hypertext Transfer Protocol
  ⊞ HTTP/1.1 302 Found\r\n
    Date: Mon, 27 Mar 2006 04:39:55 GMT\r\n
    Server: Apache/2\r\n
    Location: http://news.com.com/\r\n
    Content-Length: 204\r\n
    Keep-Alive: timeout=15, max=955\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=iso-8859-1\r\n
    \r\n
⊞ Line-based text data: text/html
  <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
  <html><head>
```

Tools and analysis methods

Processor and memory

- Commercial tools are quite heavy for production
- New JVM tools are much better
- Statistical tools are for trends, not troubleshooting
- Memory leaks/allocation issues
 - Use JVM tools such as jconsole/jrockit profiler
- Processor issues
 - Deadlocks, livelocks, overly long execution
 - Thread-dumps are your friends, but not without tools
 - Different formats/capabilities for different versions/vendors
 - See my presentation from JavaONE 2004 - *TS-1646*

Tools and analysis methods

Environmental variables

- Environment is everything not defined explicitly
 - Common interesting variables
 - OS level
 - PATH, CLASSPATH (unexpanded), JAVA_HOME, TEMP
 - JVM provided
 - Real classpath, JVM versions
 - Software provided
 - Version/Patch
 - If different variables contradict, strange things happen
 - Path may override JAVA_HOME sometimes
 - Look for environment as logged by the application
 - If that fails, various process tools show OS level info

Tools and analysis methods

Configuration files

- Common types
 - XML - parsable
 - name/value pairs (.properties) – usually parsable
 - Defaults and autowiring – problematic for maintenance
- Locations
 - Files on the filesystem
 - Registry
 - Inside the jars – hard to discover
- From our example

```
Preferences prefs = Preferences.userRoot().node("Boundaries");  
int port = prefs.getInt("port", 8001);
```

Tools and analysis methods

Configuration files

- Using RegMon (from Sysinternals)
 - Shows where java 5 preferences are kept by default
 - Notice the leading / in **/Boundaries** branch

```
javaw.exe:3760  OpenKey  HKCU\Software\JavaSoft\Prefs          SUCCESS  Access: 0x4
javaw.exe:3760  CreateKey HKCU\Software\JavaSoft\Prefs\Boundaries  SUCCESS  Access: 0x20
javaw.exe:3760  CloseKey  HKCU\Software\JavaSoft\Prefs          SUCCESS
javaw.exe:3760  CloseKey  HKCU\Software\JavaSoft\Prefs\Boundaries  SUCCESS
javaw.exe:3760  OpenKey  HKCU\Software\JavaSoft\Prefs\Boundaries  SUCCESS  Access: 0x1
javaw.exe:3760  QueryValue HKCU\Software\JavaSoft\Prefs\Boundaries\port  SUCCESS  "8004"
```

Tools and analysis methods

Configuration files

- Working with configuration files
 - Too long to read through
 - Processing extracts relevant information
 - Visualization highlights complex relationships
- Processing XML
 - XSLT/XQuery for serious use
 - XMLStarlet for prototyping
- Visualization
 - Graphvis for any $A \rightarrow B$, $B \rightarrow D$, $C \rightarrow D$ relations

Tools and analysis methods

Configuration files

- XMLStarlet – Unix style toolkit for XML
 - Example: What ports tomcat listens on?

```
... \xmlstarlet-1.0.1\xml sel -T -t
    -m //*[@port]
    -m ancestor::* -o -+ -b
    -v local-name()
    -o : -v @port
    -n
server.xml
```

```
Server:8005
```

```
++Service:
```

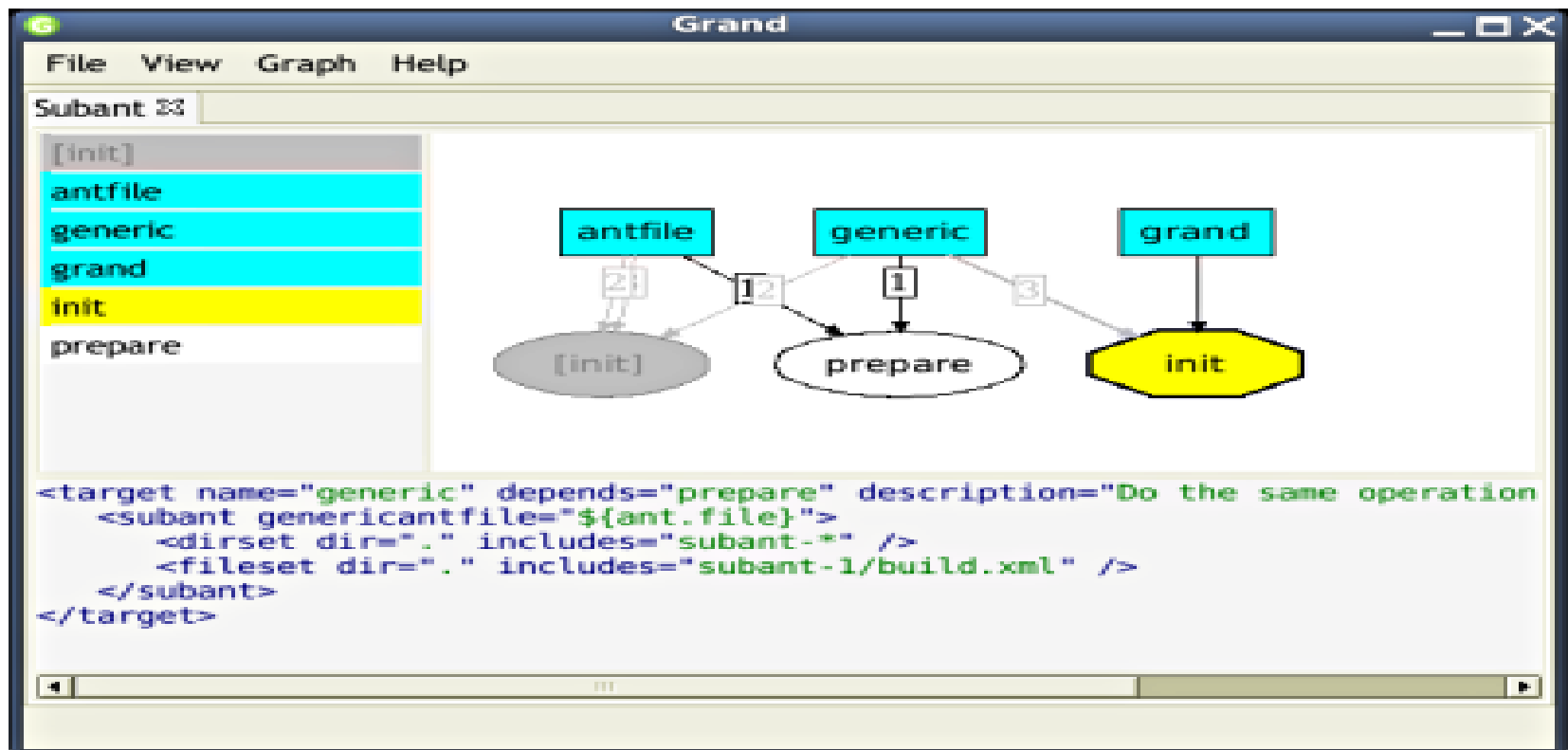
```
++-+Connector:8080
```

```
++-+Connector:8009
```


Tools and analysis methods

Configuration files

- ANT config visualisation – Grand from ggTools



Let's talk about

Why are you here? Why am I here?

Descriptive, not prescriptive approach

System boundaries and choke points

Tools and analysis methods

Proficiency across platforms

Honing the troubleshooting skills

Looking forward

Proficiency across platforms

The lazy programmer

- If you work across multiple platforms
 - Do not learn multiple tools for the same task
 - Use the same editor – Vim/Emacs
 - I use Vim + OTF (script 634) + JAD (script 446)
 - Use Unix/Cygwin tools – grep, find, sort, uniq
 - Use the same XML processor – XMLStarlet
 - Use the same network analyser – Ethereal
 - Use the same image editor – Gimp
- Do not rely solely on super-environment (Eclipse)
 - It will most probably not be installed on production

Proficiency across platforms

The lazy programmer

- Vim + OTF (On-The-Fly highlighter)

```
2006-03-22 20:05:37 StandardContext[/balancer]org.apache.webapp.balancer.BalancerFilter: init
eChain: [org.apache.webapp.balancer.RuleChain: [org.apache.webapp.balancer.rules.URLStringMat
Target string: News / Redirect URL: http://www.cnn.com], [org.apache.webapp.balancer.rules.F
arameterRule: Target param name: paramName / Target param value: paramValue / Redirect URL: h
ww.yahoo.com], [org.apache.webapp.balancer.rules.AcceptEverythingRule: Redirect URL: http://:
apache.org]]
```

```
2006-03-22 20:05:37 StandardContext[/jsp-examples]ContextListener: contextInitialized()
```

```
2006-03-22 20:05:37 StandardContext[/jsp-examples]SessionListener: contextInitialized()
```

```
2006-03-22 20:05:38 StandardContext[/servlets-examples]ContextListener: contextInitialized()
```

```
2006-03-22 20:05:38 StandardContext[/servlets-examples]SessionListener: contextInitialized()
```

Let's talk about

Why are you here? Why am I here?

Descriptive, not prescriptive approach

System boundaries and choke points

Tools and analysis methods

Proficiency across platforms

Honing the troubleshooting skills

Looking forward

Honing the troubleshooting skills

Get it before it gets you

- Do you know what you run
 - Confirm installed software version from log files
 - Find where **all** the configuration files are
 - Find where **all** the log files go to
 - Find out what the server does every 5 minutes
- Try flying blind
 - Deploy a program the normal way, *then*
 - Change a class and redeploy without restarting the server using tools installed in production **only**
- Read (not skim) and understand a log file
 - Tools like Splunk and Apache Chainsaw may help

Let's talk about

Why are you here? Why am I here?

Descriptive, not prescriptive approach

System boundaries and choke points

Tools and analysis methods

Proficiency across platforms

Honing the troubleshooting skills

Looking forward

Looking forward

Will it get easier?

- Things will get easier
 - JVMs/OSs become more instrumentable
 - New commercial and open sources products appear
 - Splunk, Apache Chainsaw, Ethereal
- Things will get harder
 - Multiple processors – more synchronization problems
 - AJAX – control is no longer in one place
 - Synchronization is now on the client
 - Requests may or may not complete correctly
 - Browsers are different
 - SOA makes everything more distributed
 - More configuration, more log files, harder to troubleshoot

Summary

- Don't panic
- Remember the 5 boundary types
- Identify which boundary/choke point may have the answer
- Know the tools and how to use them
- Harmonize tools across all platforms
- Practice beforehand
- Share the knowledge

For More Information

- Articles
 - <http://blogicblog.blogspot.com> - my blog on this topic
- Tools
 - Sysinternals: <http://www.sysinternals.com/>
 - Vim: <http://www.vim.org>
 - Ethereal: <http://www.ethereal.com/>
 - XMLStarlet: <http://xmlstar.sourceforge.net/>
 - Graphviz: <http://www.graphviz.org/>
 - Apache Chainsaw:
<http://logging.apache.org/log4j/docs/chainsaw.html>
 - Splunk: <http://www.splunk.com/> (commercial)

Q&A

Alexandre Rafalovitch

Paradise lost: Troubleshooting Java™ applications outside of development sandbox

Alexandre Rafalovitch
arafalov@gmail.com

May 2006