



# JTcl and Swank

What's new with Tcl and Tk on the JVM

Bruce A. Johnson, Tom Poindexter, & Dan Bodoh



# JTcl and Swank

## Bruce's Motivation

Cross-platform, scriptable, desktop  
applications for analyzing and visualizing large  
(multi-Gbyte) data sets

Complex applications for small market  
requires high productivity

# Why Java

- Extensive library
- Threading
- Write Once, Run on Mac, Windows, Linux...
- Stupid mistakes hard(er) to make
- More productivity, for the same reasons Tcl more productive tool than C.



# Why Java

- configure!!!!
  - Tcl 8.5.8 configure is 20162 lines long
  - Not a problem for pure Tcl, but as soon as you start using C coded extensions...
  - 32 bit/64 bit, Linux (which Linux), Mac, Windows (oops, not configure on Windows)

# Why Java

```
% set ds [java::new org.apache.commons.math.stat.descriptive.DescriptiveStatistics]  
java0x259  
% $ds addValue 9  
% $ds addValue 10  
% $ds addValue 11  
% $ds addValue 8  
% set mean [$ds getMean]  
9.5  
% set variance [$ds getVariance]  
1.66666666666667
```

# Jacl

- Java Command Language
  - Implementation of Tcl that runs on the Java Virtual Machine
- Project started by Ioi Lam and Brian Smith at Cornell (Tcl/Tk Conference 1997)
- Continued at Sun, then Scriptics, then largely maintained by Mo DeJong



# Jacl

- Part of the Tcl/Java Project
  - <http://tcljava.sourceforge.net>
  - Includes TclBlend which allows one to load the JVM into the C version of Tcl
- Most recent release is 1.4.1 in April 2008
- Essentially Tcl 8.0 w/o Tcl Byte Code Compiler
- Includes TJC - Tcl to Java Compiler & ITcl

# Jacl → JTcl

- Started with the Google Summer of Code-2009
  - Goal to bring Jacl to Tcl 8.4 (Radoslaw Szulgo)
  - After GSC we decided to fork Jacl to JTcl to accelerate the incorporation of the changes and do further modernization



# JTcd

- Drop TclBlend
  - No interest to team, makes build environment more complex

# JTcl Modernization

- Testing
  - Drive from JUnit
  - Drop platform specific tests
  - Fix erroneous test results

# JTcl Code Modernization

- Rearrange packages
- Reformat code to “standard” Java style
- Switch from make to Maven
- IDE/mvn friendly layout
- Move from 4 jars (jacl,tcljava,itcl, tjc) to 1 jar file (and it includes much of tcllib).
  - `java -jar jtcl.jar`
  - `jtcl` and `jtcl.bat`



# JTcl Code Modernization

- Regular Expressions
  - `java.lang.regex.Pattern`
  - `java.lang.regex.Matcher`
  - plus translations from Tcl to Java syntax

# JTcl Pipelines

- Pipelines for “exec” and “open” commands now supported
- Pure Java solution using `java.lang.Process` and `java.lang.ProcessBuilder`
- Redirection handled by JTcl, some limitations that may go away with Java 7

# JTcl Channels

Java Class	Description
<b>SeekableChannel</b>	Abstract class that adds seek() and tell()
<b>FileChannel</b>	Extends <b>SeekableChannel</b> to implement file I/O
<b>ResourceChannel</b>	Implements reading of a Java resource using a “resource:” prefix on the file name
<b>ReadInputStreamChannel</b>	Bridges a Tcl channel to a Java <b>InputStream</b>
<b>AbstractSocketChannel</b>	Abstract class that has common code for socket channels
<b>ServerSocketChannel</b>	Implements Tcl server sockets
<b>SocketChannel</b>	Implements Tcl sockets
<b>TclByteArrayChannel</b>	Used internally to bridge Tcl channels to Tcl byte arrays



# JTcl fcopy & File Events

- fcopy
  - Copies bytes from one channel to another
  - Copy occurs in second Java thread
- filevents
  - Depends on new non-blocking IO in channel system

# JTcl File Events

Hallmark of Tcl is the event system that allows writing servers with a minimum of code

Simple test of File Events and Channels  
Dustmote ( <http://wiki.tcl.tk/4333> )  
Web server in ~40 lines of code

It works, and spawns threads proportional to the number of simultaneous client requests.

# JTcl Summary

- ~Tcl 8.4
- + dict and apply from Tcl 8.5 (thanks to Neil Madden)



# Swank

- Success of Tcl, has a lot to do with Tk
- Swank written to provide “Tk” companion to Jacl (now JTcl) and necessary for my applications.
- Swank generated as a combination of Java code that is hand written and generated with JTcl scripts.

# Swank

Swing - Tk Correspondence - Standard Widgets			
Swing	Tk	Swing	Tk
JButton	button	JRadioButtonMenuItem	radiobutton (on menus)
JCheckBox	checkbutton	JScrollBar	scrollbar
JFrame	toplevel	JSlider	scale
JLabel	label	JSpinner	spinbox
JList	listbox	JTextArea	message
JMenu	menu	JTextField	entry
JMenuBar	menubar	JTextPane	text
JPanel	frame	JFrame (composite)	labelframe
JRadioButton	radiobutton	JPanel (customized)	canvas

# Swank

Swing - Tk Correspondence - Other Widgets			
Swing	Tk	Swing	Tk
JDesktopPane	jdesktoppane	JProgressBar	jprogressbar
JComboBox	jcombobox	JScrollPane	jscrollpane
JDialog	jdialog	JSplitPane	panedwindow
JEditorPane	html	JTabbedPane	jtabbedpane
JInternalFrame	JInternalframe	JTable	jtable
JOptionPane	joptionpane	JToolBar	jtoolbar
JPasswordField	jpasswordfield	JTree	jtree
JPopupMenu	jpopupmenu	JWindow	jwindow



# Swank Canvas

- Additional Canvas Items
  - htext: Displays basic HTML
  - connector: Line between two other objects
  - annotation: Arrow with text at other end
  - charts: Implemented with JFreeChart
- Affine Transforms
  - Transform configuration for each item
  - Custom items can generate their own
  - Canvas wide allows zooming whole canvas

# Swank Canvas

- Item Handles
  - Intrinsic aspect of each item
  - handle subcommand
- Scene graph
  - Each item has a -node option
  - New canvas node item
  - Traditional canvas is scene graph with one root
  - raise/lower operate on nodes

# Swank Canvas3D

- Rendering with Java 3D
- Create items like as on traditional canvas, but with x,y,z coordinates
- Sphere, Cylinders, Cones, Text
- I use custom 3D molecule item



# Swank

## What's Next - TkFX

- Swank depends on Java Swing
  - Swing is unlikely to see further development by Oracle
- JavaFX is the new focus of Oracle for Java GUI
  - New hardware accelerated graphics engine (Prism)
  - New windowing toolkit (Glass)
  - Media Engine for streaming content
  - Web component based on WebKit
  - 3D graphics

# TkFX



toplevel  
button  
entry  
grid  
raise  
lower  
wm  
bind  
webview  
htmleditor



# JTcl & Swank

- JTcl
  - <http://jtcl.kenai.com>
- Swank
  - <http://swank.kenai.com>
- Veery
  - <http://veery.kenai.com>
  - (Vector data inspired by Vlerq/Metakit in Java 0.0)
- Hyde
  - <http://aejaks.sourceforge.net>
  - Critcl for Java (generate and compile Java code from JTcl)