VA Smalltalk Update

John O’Keefe
Chief Technical Officer
Instantiations, Inc.
Agenda

• Company Update
• Current Release Content
• Next Release Candidates
• Q&A
Company Updates
VAST remains strong - new technology and users

- Intense new technology development
  - Upcoming product release
  - New 32/64-bit VM
- Community Outreach
  - Conference/workshop sponsors and participants
    - ESUG, FAST Smalltalks
- Sponsored Marquette Camp Smalltalk
  - September 15 – 18, 2016
- Hosting North Carolina Camp Smalltalk
  - Spring 2017
2015-2016 Engineering Focus

- Enhanced Cryptography Support
- TCP IPv6 Support
- Fast Reliable Headless Installation
- Build System Re-engineering
- New 32/64-bit VM
New Release

V8.6.3 available Q4 2016!
V8.6.3 Content
Base Class Libraries

- Zip/Unzip support
  - Inflate/deflate convenience APIs added
- OSLong class added
  - Simplifies handing of 64-bit data and pointers
    - Transparent resizing on 32/64 bit systems
    - Transparent platform adaptation
Communications Enhancements

- IPv6 support
  - All layers enabled
    - SocketCommunicationsInterface
    - Server Smalltalk (SST)
    - Parts
  - UI handles IPv4 and IPv6 addressing syntax
  - .INI file preference controls addressing behavior
    - IPv4 is default
V8.6.3 Contents
Seaside and Grease

• Small currency updates
• Adapted to IPv6
V8.6.3 Contents
Cryptography

- OpenSSL 1.1 Compatibility
  - *Lots of new algorithms available*
- Continue to support/enhance 1.0.x
- Compatibility layer to handle API breakage
  - OpenSSL 1.1 is a significant overhaul
  - Good amount of API breakage from 1.0.x
  - We have handled that internally
  - No code changes required for the user to move up
Secure Memory Support

Windows
- User can request bytes that are encrypted in-memory
- Auto-decrypt-encrypt during OpenSSL native calls
- Uses Microsoft Crypto API

UNIX/Linux
- Secure Arena
- Page-guarded on either side
- Pinned to RAM (won’t swap to disk)
- Won’t show up in a core-dump
V8.6.3 Contents

SQLite

• Update SQLite to 3.15.0
  • Transparent performance improvements
  • Bug fixes
New popup menu on Environments list pane provides new actions:

- Open a file explorer on the VA Smalltalk installation folder associated with the selected Environment
- Open a command (terminal) window on the selected Environment’s folder
- Open a command (terminal) window on the VA Smalltalk installation folder associated with the selected Environment
- Duplicate the selected Environment
  - Setup and use standardized images
  - Take checkpoint of development activity
V8.6.3 Contents
Installers

- All UNIX installers are headless
  - .RPM (Fedora RedHat derivatives)
  - .DEB (Debian derivatives)
  - .PKG (Solaris)
- Windows installer can be scripted
  - Supports standard installation across multiple machines
V8.6.3 Content
New Supported Platforms

- Ubuntu 16.10
- Fedora 24
- RedHat Enterprise Linux 7
Reengineered Build System

• Old Image Build and Installation Build
  • All custom Smalltalk code
    • Dates to mid-90’s with relatively small changes since
    • Not the best code quality
  • Builds were slow with significant manual intervention
  • Windows and UNIX builds were sequential
  • Difficult to restart if problem occurred
  • Installation artifacts were unmanaged

• New Image Build and Installation Build
  • Cmake-based system used to script build
    • Can restart at any job step
    • Duplicate and redundant processing removed
  • Still uses Smalltalk function where appropriate
    • Driven using abt.cnf scripting
  • Fully-automated nightly builds (if changes occurred)
  • Installable artifacts will be managed in Git repository
Reengineered Build System (cont)

- **Old Installation**
  - Smalltalk packaged image
    - Difficult to maintain
  - Slow install
    - File-by-file copying
    - File attributes in separate shadow file

- **New Installation**
  - ‘Standard’ installers
    - Windows MSI
    - UNIX RPM/DEB/PKG
  - Documentation in separate packages
  - Smaller download packages
  - FAST install
Reengineered Build System (cont)

- Old Build Testing
  - Only VM Build testing was automated

- New Build Testing
  - Automated build testing using CMake/CTest
    - Install Verification Tests
    - VM Tests
    - Image Tests
  - All platforms can be tested in parallel
  - Currently over 10,000 mainline testcases (and growing)
    - Additional non-automated testcases for loadable features
Looking to the Future
Future Releases

• Release schedule is about once a year
  • Depends on volume of content
  • Current content information in Product Roadmap

• Content based on requirements from:
  • Surveys
  • Direct customer interactions
  • Forums
  • Support cases
  • Internals
Next Release
Candidate Items

• Web interface
  • Seaside 3.x
  • Continuation support

• Middleware
  • Postgres
  • NOSql (Voyage/MongoDB, Riak, Cassandra, or ???)
Next Release
Candidate Items

- GUI Look-and-Feel
  - Common Widgets Application Window framework
    - Useful for SUnit, Seaside, etc.
  - Windows Common Controls
    - TreeView improvements
  - GTK to replace Motif on UNIX platforms

- Communications
  - HTTP/2
  - 0MQ

- Server
  - Easy to use server farm support
Next Release
Candidate Items

• Development Tools
  • Improved code library access over WAN
  • Revamped Changes Browser
  • New Code Merge Engine

• Performance and Scalability
  • Incremental garbage collection
  • 64-bit Smalltalk
Dino2 32/64 Bit VM Project
Overview

• Project Goals
  • 64-bit VMs for x86, PowerPC and SPARC
  • 32-bit VMs with performance at least as good as current production VMs
  • Improve build systems and testing infrastructure

• Production VM Review
  • Proprietary Smalltalk Model (generates assembly)
  • VM Interpreter/JIT/Primitives are generated assembly
  • Supporting Modules written in C
  • @135,000 lines of ASM
  • @50,000 lines of C
Dino2 32/64 Bit VM Project
Current Status

• Build/Compiler Infrastructure
  • CMake-Based Build System
  • GCC, MinGW and MSVC compilers

• 32/64-bit Virtual Machine
  • Running on Windows and Linux
  • Current Focus: Interpreter Performance

• 32/64-bit Smalltalk Image
  • 32-bit -> 64-bit Image Translation Complete
  • Core Smalltalk Image and many libraries are 64-bit prepped
  • Current Focus: Continued 64-bit library prep
Dino2 32/64 Bit VM Project
The Journey Forward

- **Raptor**
  - 1st Generation C-Interpreter
  - Slow but 64-bit Clean
  - 80% bytecode speed (32-bit)
  - 50% message send speed (32-bit)
  - Primitive call machinery slow
  - Smalltalk process switching slow
  - Primitive implementation often faster than production
  - Allowed us to move forward with Image work
  - @90,000 lines of C code
Dino2 32/64 Bit VM Project
The Journey Forward

- Indominus-Rex
  - 2nd Generation C-Interpreter
  - Faster C-Implementation
  - 85% bytecode speed (32-bit)
  - 75% message send speed (32-bit)
  - Primitive implementation often faster than production
  - Stable - reference implementation for new platforms
  - Small changes to large interpreter loop resulted in unpredictable behavior
    - Register allocator having a difficult time
    - Constant fight with the compiler
  - @85,000 lines of C Code
Dino2 32/64 Bit VM Project
The Journey Forward

• Coelo
  • LLVM Code-Generated Interpreter
    • Compiler toolkit
    • SSA Abstract Assembly Representation
  • Still down one register on X86 compared to production VM
    • Can’t use hardware-stack register (ESP)
    • Superior Code-Gen makes up for it
  • 100% bytecode speed (32-bit)
  • 110% message send speed (32-bit)
  • At least 20% performance jump in primitives
  • For many prims (Floats) the production VM used call-outs to C
    • These are described directly in LLVM
    • Speedups are more like 4-6x
  • @19,000 of C++ Code (Interpreter Code-Gen)
  • @75,000 of C Code
Dino2 32/64 Bit VM Project

Customer Involvement

- Early Customer Access Program (ECAP)
  - Kicked off our ECAP program in July, 2016
  - Select customer involvement
  - Opportunity for feedback and collaboration
How Do You Get VA Smalltalk?

- Download evaluation copy
- Buy development licenses
- Download development build
  - Announced in VA Smalltalk Google Group
- Be a committer on an Open Source project
  - http://www.instantiations.com/company/open-source.html
- Work for an educational institution
  - http://www.instantiations.com/products/academic-license-program.html
Contact us

• General information
  • info@instantiations.com

• Sales
  • sales@instantiations.com

• Support
  • support@instantiations.com

• Me
  • john_okeefe@instantiations.com
Thank you for your attention

Questions?
Backup
Future Releases
Candidate Items

- Web interface
  - Web services debugging support/doc
  - Web services tooling improvements
  - Validating XML parser

- GUI Look-and-Feel
  - GTK+ 3.x on Linux
  - Windows Common Controls additions
  - Back-port widgets from add-ons (ex: UML Designer)
Future Releases
Candidate Items

- **Infrastructure**
  - Consolidate product settings using Settings Framework
  - Settings Dialogs to complement Settings Framework
  - Consolidate product logging using Logging Framework

- **Middleware**
  - GLORP Programmer’s Reference
  - MQ currency
  - TCP/IP V6
  - 0MQ
Future Releases
Candidate Items

• Development Tools
  • Redesigned Change Browser & Merge Tool
  • C API wrapper generation tool

• Security
  • ‘Full’ security framework -- OpenSSL 1.0.2 or later wrappers

• Performance and Scalability
  • Parallel garbage collection
  • Class library performance hotspots
  • Integrate KES/Stats goodie for object monitoring

• Platform
  • Windows Services control moved from C to Smalltalk
Future Releases
Candidate Items

• External Interfaces
  • JNIport (currently available on VAStGoodies.com)
  • .NET/C#
  • Other

• Class Libraries
  • Collection hashing policies
  • Sorted collection sorting policies

• Other
  • *We’re always looking for suggestions*
Announcements

- A framework for event notification
  - Based on Observer pattern
  - Replaces traditional event symbols (#changed) with event objects (ChangedEvent)
  - Event objects can carry context-oriented information
  - Clients can subscribe, unsubscribe, and handle event
    - Event handling uses #on:do: like exception handling
- Supplied because Roassal uses it, but you can use it too
Roassal Visualization Framework

- Roassal is an open-source project of ObjectProfile
- Roassal graphically renders objects using short expressive Smalltalk expressions
- A large set of interaction are offered for a better user experience
  - Painting
  - Brushing
  - Interconnecting
  - Zooming
  - Dragging and dropping
- Using Roassal can enhance your understanding of any arbitrary object model
Roassal Visualization Framework
continued
Open Source Project Committer

- Instantiations will donate a perpetual VA Smalltalk license
  - to committers
  - working on non-commercial open source project
  - Hint: release something to VAStGoodies.com
- **No charge**
- **Details**
  - [http://www.instantiations.com/company/open-source.html](http://www.instantiations.com/company/open-source.html)
Academic Licenses

- Accredited educational institutions
  - Teaching staff
  - Students
- **No charge**
- Details
  - http://www.instantiations.com/products/academic-license-program.html