



**JavaBeans™:
New Business Opportunities with
Java™ Components**



Frank Rimalovski
Sr. Product Manager, JavaBeans
Sun Microsystems, Inc.

Today's Speakers

- ◆ **Tom Kim**

- ◆ Product Manager, Java Products
Rogue Wave Software, Inc.

- ◆ **George Kassabgi**

- ◆ Java Engineering Manager/Technologist
Progress Software Corp.

- ◆ **Dean Guida**

- ◆ President and CEO
ProtoView Development Corp.

- ◆ **Peter Cohen**

- ◆ Marketing Manager
Lotus Development Corp.



Why JavaBeans?

- ◆ JDK™ 1.0
 - ◆ Object orientation
 - ◆ Portability
 - ◆ Network aware
 - ◆ etc.
- ◆ What about?
 - ◆ Object persistence
 - ◆ Dynamic applet interaction
 - ◆ Support for visual assembly

What Is JavaBeans?

- ◆ *The* component model for Java
 - ◆ Reusable
 - ◆ Portable
 - ◆ Interoperable
 - ◆ Secure



How Did We Get Here?

- ◆ May '96: JavaOne
- ◆ Oct '96: Froze JavaBeans 1.0 spec
- ◆ Dec '96: BDK 1.0 beta
- ◆ Feb '97: JavaBeans core to JDK 1.1
BDK 1.0 FCS
JavaBeans Bridge for
ActiveX beta



How Did We Get Here?

- ◆ Developed with many partners
 - ◆ Apple, Baan, Borland, CI Labs, Corel, Informix, IBM, JUSTSYSTEM, Lotus, Microsoft, Netscape, Novell, Oracle, ParcPlace, Silicon Graphics, SunSoft, Sybase, Symantec, Texas Instruments, Visual Edge...
 - ◆ Plus many external reviewers

Strategy for JavaBeans

- ◆ Preserve portability and reusability
- ◆ Keep it simple and lightweight
- ◆ Exploit strengths of Java language
- ◆ Platform integration
- ◆ Core part of Java platform



Rogue Wave Software
Tom Kim
Product Manager

Software Development Trends

- ◆ Buy vs. build
 - ◆ Recognition that it's cheaper to buy non-core technologies
 - ◆ Adoption of object oriented design/programming
 - ◆ Reuse of software components/libraries
 - ◆ Faster cycle time
 - ◆ Better quality of code

Software Development Trends

- ◆ Software increasingly viewed as resource to be used vs. being produced

Rogue Wave Software

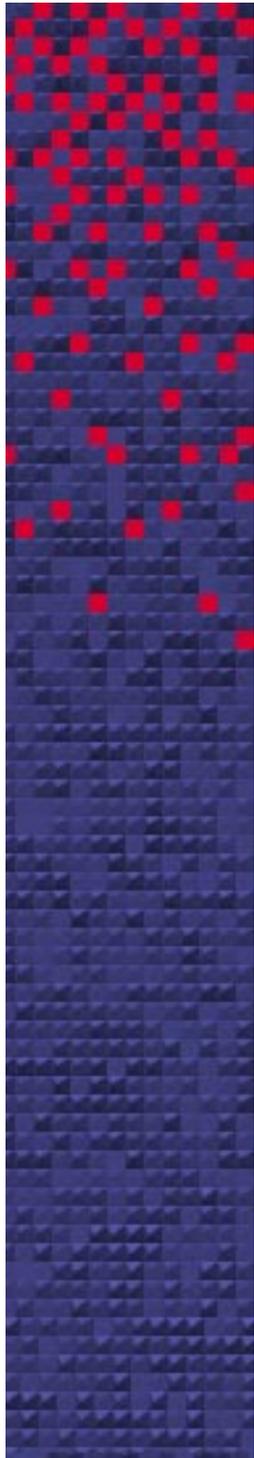
The Software Parts Company

- ◆ Mission Statement
 - ◆ Manufacture reusable software parts
 - ◆ Make our customers successful at using these parts
 - ◆ Make our customers successful at building their own parts



Projects and Products

- ◆ JWidgets
 - ◆ Taking about 3.5 days per widget to move from a 1.02 class library to a Bean
- ◆ JChart
- ◆ JDBCTools



Demo



Why JavaBeans?

- ◆ Easy integration into IDEs via the Beans Builders
 - ◆ AD tools integration via JavaBeans and customizer facet of JavaBeans
- ◆ Broadening of target market
 - ◆ Components accessed via traditional API
 - ◆ Components accessed via customer via IDE

Business Issues

- ◆ Packaging
 - ◆ By component (Bean), or
 - ◆ By bundling multiple Beans like a library
- ◆ Licensing
 - ◆ Per developer per platform, or
 - ◆ Per developer

Business Issues

- ◆ Pricing
 - ◆ If sold Bean-by-Bean, at low prices, how to price support
- ◆ Source or byte code
 - ◆ Does selling source code make sense for Java and Beans as in C++

Summary

- ◆ Java is being viewed as a serious technology
- ◆ JavaBeans is a good component model
- ◆ A lot of uncertainty and risk in Java market
- ◆ No choice but to jump in if we want to be a contender



JavaOneSM
Sun's 1997 Worldwide Java Developer Conference

Progress Software
George Kassabgi
**Java Engineering Manager/
Technologist**

Who Are We?

Progress Software (NASDAQ:PRGS)

- ◆ 4GL/Database/Tools Vendor
 - ◆ PROGRESS V8
 - ◆ PROGRESS Database/
DataServers
 - ◆ PROGRESS SmartObjects
 - ◆ WebSpeed
 - ◆ Crescent for VB, J++ Developers
- ◆ Focus on VAR business

Progress and JavaBeans

- ◆ SmartObjects technology using Java and JavaBeans architecture
- ◆ Methodology for building business applications “enterprise class”
- ◆ Using “open” solution
 - ◆ Choice of JavaBeans IDEs
 - ◆ Integration with other Beans

Why JavaBeans?

Our view on components

- ◆ Business application developers
 - ◆ Need *useful* components
 - ◆ Strong component tool support
 - ◆ Extendibility
 - ◆ Tired of proprietary ‘solutions’
 - ◆ Need platform independence
 - ◆ View Internet as decision *catalyst*

Why JavaBeans?

Useful components

- ◆ Strong built-in business application functionality/purpose
 - ◆ JavaBeans architecture +
 - ◆ Database savvy
 - ◆ UI framework
 - ◆ ‘Distributability’
 - ◆ Event model
 - ◆ Goes beyond source/listener model

Why JavaBeans?

Levels of abstraction

- ◆ Relative to business applications
 - ◆ JavaBeans is only the beginning
 - ◆ Need to add layers of value
- ◆ Focus on: Business developers
 - ◆ Used to 4GL languages
 - ◆ Used to drag and drop tools
 - ◆ Want everything yesterday!

Why JavaBeans?

Components don't stand still!

- ◆ Platform independence story
 - ◆ UNIX and Mac are important
 - ◆ Internet is catalyst to multi-platform
 - ◆ Better safe than sorry
- ◆ Components ready to travel
 - ◆ Easily transportable and deployable
 - ◆ Small footprint

Business Opportunities

JavaBeans 'on the move'

- ◆ Selling solutions oriented around:
 - ◆ JavaBeans component frameworks
 - ◆ Server side business logic
 - ◆ Object persistence
- ◆ Concentrate on the 'back-end'
 - ◆ Businesses are run on the server
 - ◆ Server side JavaBeans

Business Opportunities

JavaBeans evolution

- ◆ Simple UI controls
- ◆ Data aware controls
- ◆ Component suites
- ◆ Component frameworks
- ◆ ...and beyond
- ◆ Stay ahead of the game

Marketing Beans

Use real-world solution examples

- ◆ Companies building Java today
- ◆ Need case studies, success stories
- ◆ Show component leverage
- ◆ Don't get lost in the crowd

Partnerships

Think of layering

- ◆ ‘Wrap’ other vendor’s Beans
- ◆ Example:
 - ◆ An intelligent business application grid presentation component
 - ◆ Uses a third party GRID.class
 - ◆ Includes third party JDBC stubs
 - ◆ May use calls to Java through ORB

Partnerships

Same challenges, different day...

- ◆ Many aspect of building a business application remain the same
- ◆ Partner with current leaders in:
 - ◆ Data storage
 - ◆ Modeling
 - ◆ Services
 - ◆ World wide support

Summary

JavaBeans = Component Story

- ◆ ‘The night is young’
 - ◆ IDE vendors are catching up
 - ◆ ActiveX bridge changes the game
 - ◆ Experience with BDK has just begun
- ◆ Choose component strategy
- ◆ Design today, for tomorrow’s component requirements



ProtoView Development

Dean Guida
President and CEO

ProtoView and JavaBeans

- ◆ DataTableJ
 - ◆ Built for maximum speed and performance
 - ◆ Data class separate from View class
- ◆ CalendarJ
 - ◆ Rich display features
 - ◆ 1,3,6 and 12 month displays

ProtoView and JavaBeans

- ◆ TreeViewJ
 - ◆ Unlimited nodes
 - ◆ Sorting, searching, colors and fonts
- ◆ WinJ
 - ◆ Collection of UI components
 - ◆ Mask, Edit, Currency, Numeric, Date, Time, Round button, Button, Text3D
 - ◆ Rich collection of display options

Why JavaBeans?

- ◆ Delivers on the promise of object reuse
- ◆ Simply and elegant component architecture
- ◆ Leverage current C++ and OOP skills

Why JavaBeans?

- ◆ Leverage R&D expense
 - ◆ Sell to Windows, Unix, Mac and many others coming
 - ◆ Other devices: NC, PDAs
- ◆ Works in ActiveX containers
- ◆ Network ready

Market Opportunities

- ◆ Emerging market
- ◆ No declared market leader
- ◆ Wide open market

Market Challenges

- ◆ Rise above the noise
 - ◆ Product positioning
 - ◆ Clear added value message
- ◆ Distribution
 - ◆ Selling online
 - ◆ Channel conflict
 - ◆ Traditional channel distribution

Market Challenges

- ◆ Fast pace of innovation
- ◆ Maintaining margins to fund growth and profits
- ◆ Need to support components in multiple OS and hardware
- ◆ Need to add a digital marketing plan to the marketing mix

Licensing Issues

- ◆ Per developer seat
 - ◆ Price sensitive market
 - ◆ Highly competitive
- ◆ Per server
 - ◆ Public Internet access
 - ◆ Server side component

Licensing Issues

- ◆ Per client machine
 - ◆ Internal corporate system
- ◆ Per usage
 - ◆ Deliver of unique content

Partnering Strategies

- ◆ Its never fun to go it alone
- ◆ Leverage each others resources
- ◆ Obtain more market and mind share
- ◆ Typically the cost of doing business is time...not cash
- ◆ Share each others list for direct marketing activities



Lotus Development
Peter Cohen
Marketing Manager



Frank Rimalovski
Sr. Product Manager, JavaBeans
Sun Microsystems, Inc.

How to Reach...

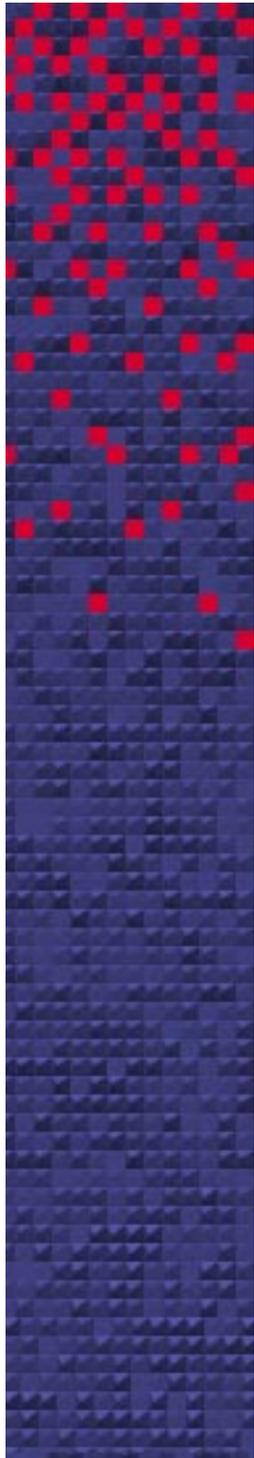
- ◆ Rogue Wave Software (#800)
 - ◆ www.roguewave.com
- ◆ Progress Software
 - ◆ www.progress.com
- ◆ Protoview Development (#206)
 - ◆ www.protoview.com
- ◆ Lotus Development (#1220)
 - ◆ kona.lotus.com

JavaBeans Sessions

- ◆ JavaBeans and the Enterprise
 - ◆ Thursday, 12:30 - 1:30, Hall D-2
- ◆ Introduction to JavaBeans
 - ◆ Thursday, 3:15 - 4:15, Esplanade
- ◆ JavaBeans Basics
 - ◆ Thursday, 4:30 - 5:00, Esplanade
- ◆ Advanced JavaBeans Development
 - ◆ Friday, 4:15 - 5:15, Esplanade

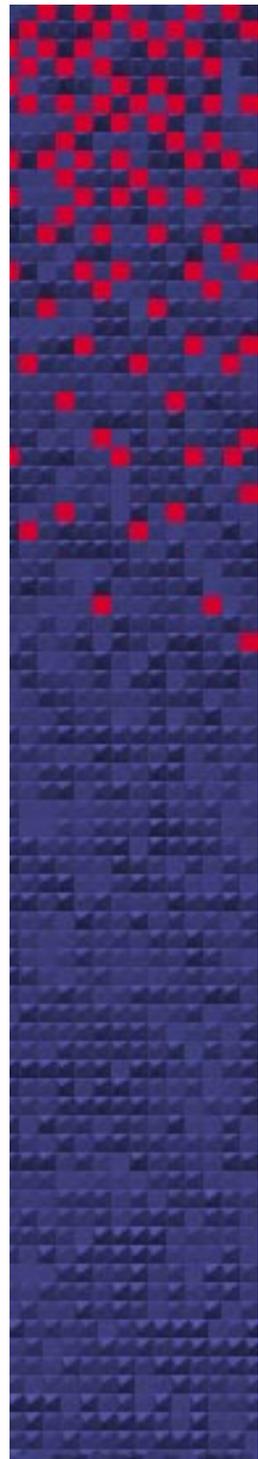
Other Valuable Sessions

- ◆ Java in the Enterprise
 - ◆ Wednesday, 1:30 - 2:30, Room 103/104
- ◆ Java Advanced Programming Tutorial
 - ◆ Wednesday, 5:00 - 6:00, Esplanade
- ◆ AWT: The Next Generation
 - ◆ Thursday, 10:00 - 11:00, Esplanade
- ◆ Distributed Computing with Java RMI
 - ◆ Friday, 10:00 - 11:00, Esplanade
- ◆ AWT Advanced Tutorial
 - ◆ Friday, 3:00 - 4:00, Esplanade



Q & A





JavaOneSM

Sun's 1997 Worldwide Java Developer Conference™