

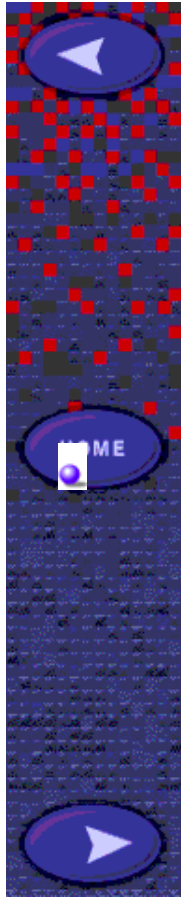


JavaBeans™ Components Architecture

Graham Hamilton

Sun Microsystems, Inc.

[Other Available Formats](#) | [JavaOne Home Page](#)



JavaBeans™ APIs

The JavaBeans technology is the component model for the Java™ Platform

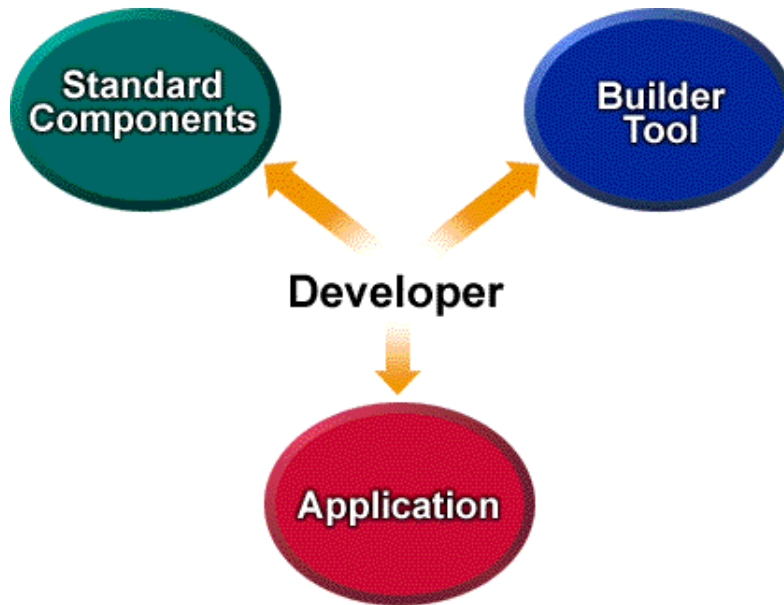
It defines Java software components

And how they fit together

Other Available Formats



What We're Enabling



Other Available Formats



What's a Bean?

**"A JavaBeans component is a
reusable software component
that can be visually manipulated
in builder tools."**

Other Available Formats



Kinds of Beans

Buttons, sliders, GUI controls

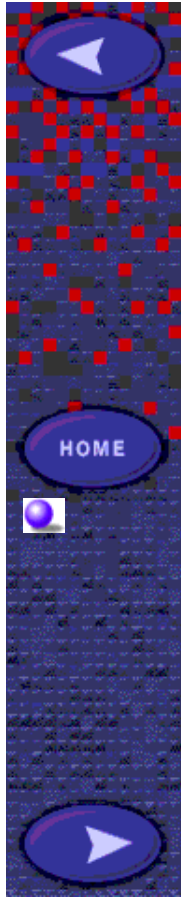
Database viewers, stock data feeds

In-house customizable mini-applications

Word processors, spreadsheets

Invisible server beans

Other Available Formats



Kinds of builders

Simple GUI layout managers

Web page builders

Full-scale application environments

Builders for invisible server applications

Other Available Formats



Platform integration

Have to integrate well with ActiveX/COM
Beans must work well in COM containers
Including Visual Basic, Delphi, IE, Word, ..
Integration in Navigator is very important
Plus normal window system integration
X11, win32, Mac, whatever
But still run everywhere!

Other Available Formats



Key Strategies

**Make beans look simple
But allow complexity**

Exploit Java language features

Make beans usable without tools

**Make people productive quickly
And keep a smooth learning curve**

Other Available Formats



Key Technologies

Events
Properties
Introspection
Customization
Persistence
Distributed systems

Other Available Formats



Events

- AWT team has designed a new event model
 - We've adopted it for beans
- Goals were:
 - Make it easier to connect methods to events
 - Use strong typing
 - Use standard "design patterns"
- When a bean fires an event
 - It invokes a named, typed method
 - On a named, typed interface
- Bean must allow registration of listeners

Other Available Formats



Properties

- **Properties are chunks of a bean's state**
 - **E.g. background, font, price, ...**
- **Accessible via getter/setter methods**
- **"bound" => change notification**
- **"vetoable" => changes can be rejected**

Other Available Formats



Introspection

How does a tool learn about a bean?

You might use a component descriptor file

But that tends to be pretty ugly

Beans uses analysis of class methods

Applying standard design patterns

To deduce properties/events/methods

Developers can also specify them precisely

Other Available Formats



Customization

In builders you want to tailor components

We support standard "property sheets"

This is good for simple things

But awkward for complex controls

Beans also supports "Customizer" classes

These can provide "wizard" guides

Custom property editors

We supply default editors for many types

Beans may provide their own type editors

Other Available Formats



Persistence

- Beans persistence uses serialization
 - Automagic persistence
 - Using the Java virtual machine.
 - Analyzes objects, pointers, fields
- People can override default behavior
- Note: Many tools use "generated source code" to achieve persistence

Other Available Formats



Distributed Beans

CORBA

RMI

RMI over IIOP

Other Available Formats



New Beans Developments

InfoBus: For exchanging structured data

JavaBeans Activation Framework

For using beans as MIME viewers

Containment and Services Protocol

For grouping beans into contexts

Enterprise JavaBeans™ components

They support middle-tier "server" apps

Components reflect business logic

Typically are run in OLTP transactions

Other Available Formats



Major Partners

We developed beans 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, ...**

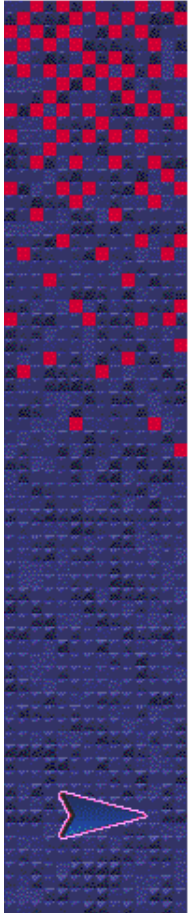
Plus many external reviewers

Many thanks to everyone who contributed!

Other Available Formats



Other Available Formats



Developing JavaBeans™ Components

Reginald Adkins

Sun Microsystems, Inc.

Other Available Formats



Overview

- **The Beans Development Kit (BDK)**
- **How To Build and Interconnect JavaBeans Components**
- **How To Create Well Behaved Bean Components**

Other Available Formats



BDK 1.0

- **BDK**

- Supports early development of Beans
- Standard reference for developers and tool vendors

- **Contents**

- The BeanBox test container
- Example Beans
- Reusable reference source code
- Makefile information
- Information located on our web site
<http://java.sun.com/beans>

Other Available Formats



Building Beans

- Programming
- Packaging
- Reuse

Other Available Formats



Programming

- **JavaBeans components are implemented in the Java™ programming language**
- **We exploit the strengths of the Java platform**
- **AWT 1.1. Event Model**
- **Standard design patterns**

Other Available Formats



Strengths of the Java™ Platform

- **Reflection API**
 - **Used for Introspection**
- **Object Serialization**
 - **Used to persistently package beans for reuse**

Other Available Formats



Packaging

- Serialized Prototypes
 - Used for pre-customization
- JAR Archives
 - Used as the standard mechanism for delivering Beans

Other Available Formats



Reuse

- **Builder tools and human programmers can easily instantiate existing beans**
- **Further customization through subclassing**
- **BeanInfo**

Other Available Formats



BeanInfo

- Crucial for reuse in builder tools
- Allows control over what properties, events and methods are exposed

Other Available Formats



Interconnecting Beans

- AWT 1.1 Event model
 - Fire events against interfaces
- What a builder may choose to interconnect
 - Properties
 - Methods

Other Available Formats



Interconnecting Beans

- **Interconnecting Properties**
 - **Target beans can veto a pending property change, and/or be notified of a completed change**
- **Interconnecting Methods**
 - **Events sources can invoke listener methods directly, or**
 - **Adaptors can be used to invoke arbitrary (zero-arg) target methods**

Other Available Formats



Interconnection Types

- **Adaptors**
- **Persistent connections**

Other Available Formats



Tips

- Follow the design patterns
- You may already be following the design conventions!
- Provide associated BeanInfo classes

Other Available Formats



JavaBean Sessions

- **Advanced topics in JavaBeans components**
Hall B, Thursday 2:45 pm
- **JavaBeans components & ActiveX/COM**
Hall B, Friday 2:45 pm
- **InfoBus: Data exchange for the entire family**
Esplanade, Friday 12:15 pm
- **JavaBeans, New Technologies Discussion**
Room 220, Friday 1:30 pm

Other Available Formats