JavaBeans

- Formally introduced alongside Java 1.1
  - Bean Development Kit (BDK) 1.0
- Impetus for many of the new features of Java
  - inner classes
  - introspection
    - the Reflection class
- Also involves:
  - serialization
  - JAR files
  - etc.

Transentia Pty. Ltd. DonationWare

JavaBeans

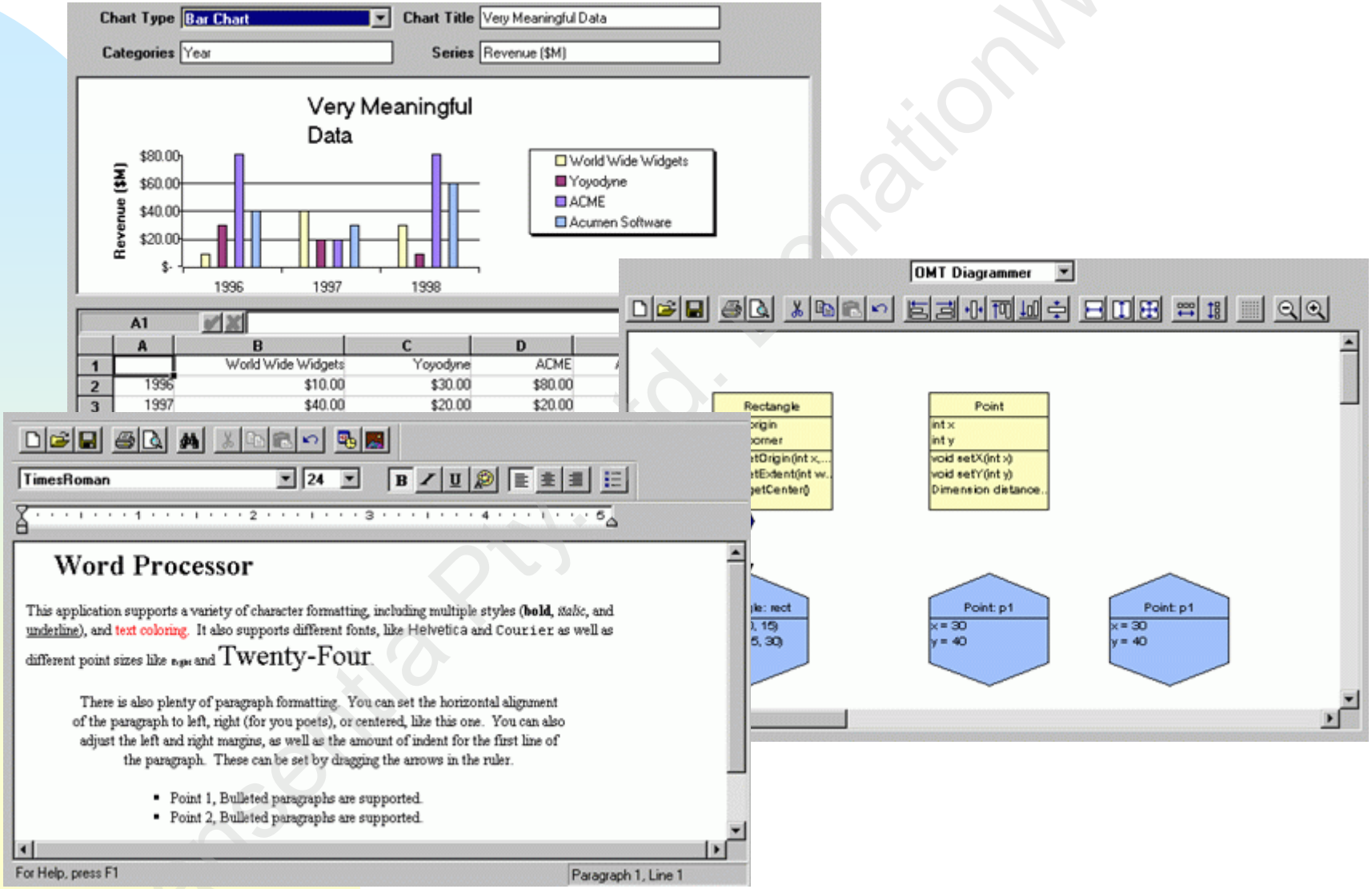
- A bean is:

  "A reusable software component that can be manipulated visually in a builder tool"

  - Not necessarily a visual thing itself
  - Not just a widget
    - a mini-application building block, possibly highly customizable
  - five common attributes: introspection, properties, events, customization, persistence
  - part of a larger component architecture that is evolving (and growing larger) all the time
- Another bullet in the war against ActiveX
  - designed to allow interoperability

Transeria Pty Ltd Donation Ware

JavaBeans



Transcella Pty. Ltd. - NationWare

JavaBeans

- Beans can be used at three levels:
  - bean-aware tool developers
    - beans API targeted mainly at this group
  - bean author
    - needs to implement a subset of the API relevant to the bean being built
  - bean user
    - don't need to use the API at all, (perhaps) only the visual builder tool
    - domain experts assembling from a palette of tools

Transentia Pty. Ltd. DomainWare

JavaBeans

- Why JavaBeans?
  - reuse
  - ease of development
    - little more difficult than a ‘normal’ applet
  - scripting/user development
    - but don’t believe all the hype about scripting & “snap together” application development… programming will always be *hard*
  - the internet has changed the rules regarding what a component should be like
  - “…the only architecture you should consider if you are developing in a Java environment.”

Transential Politics! Eg. DonationWare

JavaBeans

- *"A Bean is any Java class which adheres to certain property and event interface conventions."*
  - no Bean class: uses a protocol with other beans
    - a primitive, unformalised applet
  - heavy use of introspection
    - container or builder tool can determine the bean's characteristics by recognizing predefined patterns
- Protocol is discovered based on public characteristics:
  - properties, events & methods

Transentia Pty. Ltd. DonationWare

JavaBeans

- Properties
  - visible internal state
    - inspected by a builder tool or container
    - *not* instance variables
    - always accessed via setters & getters
  - simple property
    - *PropertyName* is a simple variable

```
public void setPropertyName (PropertyType)
public PropertyType getPropertyName ()
public boolean isPropertyName ()
```

  - indexed property
    - *PropertyName* is an array

```
public void setPropertyName (PropertyElement [])
public PropertyElement [] getPropertyName ()
public void setPropertyName (int, PropertyElement)
public PropertyElement getPropertyName (int)
```

a class is inspected looking for patterns like these

Transentia Pty. Ltd. DonationWare

JavaBeans

- bound property

```
public void setSalary (Float newSalary)
  {
  Float oldSalary = this.salary;
  this.salary = newSalary;
  vListeners.firePropertyChange ("salary", oldSalary, newSalary));
  }
```

  - sends a PropertyChangeEvent when updated
    - PropertyChangeEvent event
    - PropertyChangeSupport class
    - PropertyChangeListener interface
      - propertyChange method
  - events reported at bean, not property, level: need to determine event/property correspondence

Transentia Pty Ltd. DonationWare

JavaBeans

- constrained property

```
public void setSalary (Float newSalary) throws PropertyVetoException
  {
  Float oldSalary = this.salary;
  vListeners.fireVetoableChange ("salary", oldSalary, newSalary);
  this.salary = newSalary;
  pListeners.firePropertyChange ("salary", oldSalary, newSalary);
  }
```

  - as bound, but changes can be vetoed
    - VetoableChangeSupport class
    - VetoableChangeListener interface
      - vetoableChange
      - examines a PropertyChangeEvent
    - PropertyVetoException

```
public void vetoableChange (PropertyChangeEvent e)
  throws PropertyVetoException
  {
  float new = ((Float) e.getNewValue ()).floatValue (),
        old = ((Float) e.getOldValue ()).floatValue ();

  if (new > (old + (old * 0.1)))
    throw new PropertyVetoException ("Too generous!", e);
  }
```

Transentia Pty. Ltd. DonationWare

JavaBeans

- Methods
  - all public methods available for use by container
    - “pattern matching” used to determine properties, etc.
    - unless alternative view specified:
      - BeanInfo interface and SimpleBeanInfo class
        - get{Method, Property, EventSet}Descriptors ()
        - getIcon ()
        - getDefault{Property, Event}Index ()
        - etc.
  - all public methods should be synchronized
    - since may be multiple views in a container

Transeftia Pty. Ltd. DonationWare

JavaBeans

```
public class SimplestBeanInfo extends SimpleBeanInfo
  {
  private final static Class beanClass = SimplestBean.class;

  public java.awt.Image getIcon (int k)
    {
    if (k == BeanInfo.ICON_MONO_16x16 || k == BeanInfo.ICON_COLOR_16x16)
      return (loadImage (”SimplestBeanIcon16.gif"));

    else if (k == BeanInfo.ICON_MONO_32x32 || k == BeanInfo.ICON_COLOR_32x32)
      return (loadImage ("SimplestBeanIcon32.gif"));
    return null;
    }

  public PropertyDescriptor [] getPropertyDescriptors ()
    {
    try
      {
      PropertyDescriptor col = new PropertyDescriptor ("Color", beanClass);
      col.setshortDescription ("The color of this SimplestBean");
      return (new PropertyDescriptor [] { col });
      }
    catch (IntrospectionException e)
      { return (super.getPropertyDescriptors ()); }
    }

  public int getDefaultPropertyIndex () { return (0); }

  ( other code elided… )
  }
```

Transentia Pty. Ltd. DonationWare

JavaBeans

- Events
  - beans can use the standard Java 1.1 events
    - might simulate a mouse click, say
    - all beans know how to deal with these events
  - can also define custom events
    - to say “hey! I’ve just done something really good and I want you to know about it…”
      - target must want to know about this specific event in the first place (have registered a listener for the event type)

```
EEvent
EListener
public void methodName (EEvent)
public void addEListener (EListener)
public void removeEListener (EListener)
```

Transentia Pty. Ltd. DonationWare

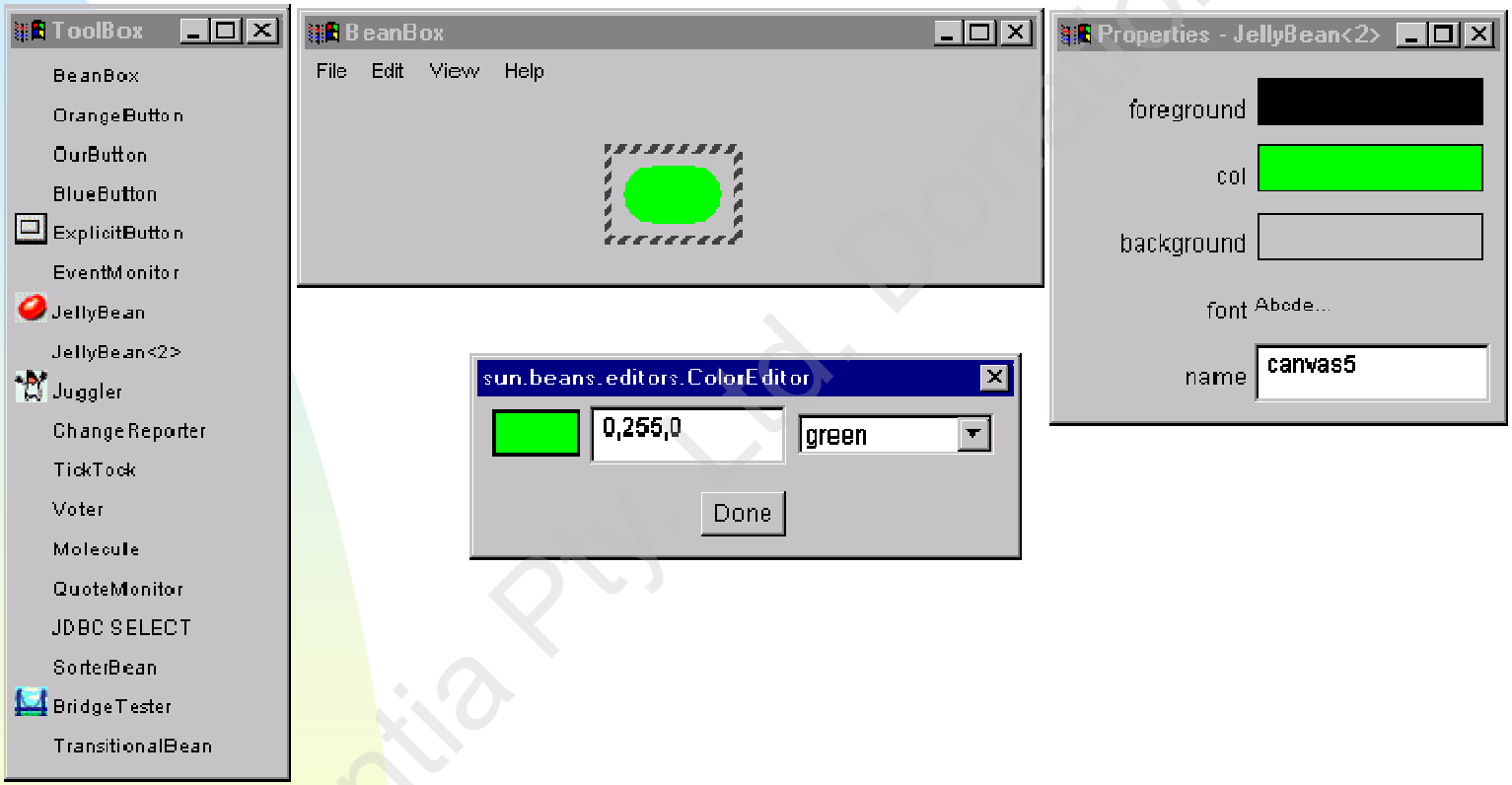
JavaBeans

```
import java.awt.*;

public class SimplestBean extends Canvas
  {
  private Color ourColor = Color.orange;

  public SimplestBean () { setSize (60, 40); }

  public void paint (Graphics g)
    {
    g.setColor (ourColor);
    g.fillArc (5, 5, 30, 30, 0, 360);
    g.fillArc (25, 5, 30, 30, 0, 360);
    g.fillRect (20, 5, 20, 30);
    }

  public synchronized Color getCol ()
    { return ourColor; }

  public synchronized void setCol (Color newColor)
    {
    ourColor = newColor;
    repaint();
    }
  }
```

Defines the bean's simple *col* property

Transentia Pty. Ltd. DonationWare

JavaBeans



Transentia Pty Ltd, Donation Ware

JavaBeans

- Property editors
  - for bean-defined, non-standard property types
- Customizers
  - allow properties to be established *en-masse* via a specialized dialog, rather than piecemeal through numerous property editors
  - may be more user (programmer) friendly

Transentia Pty. Ltd. DonationWare