WebSphere software

IBM

e business software

Java Visual Editor (JVE)

WebSphere Application Server - Express Beta

IBM Software Group

Objectives

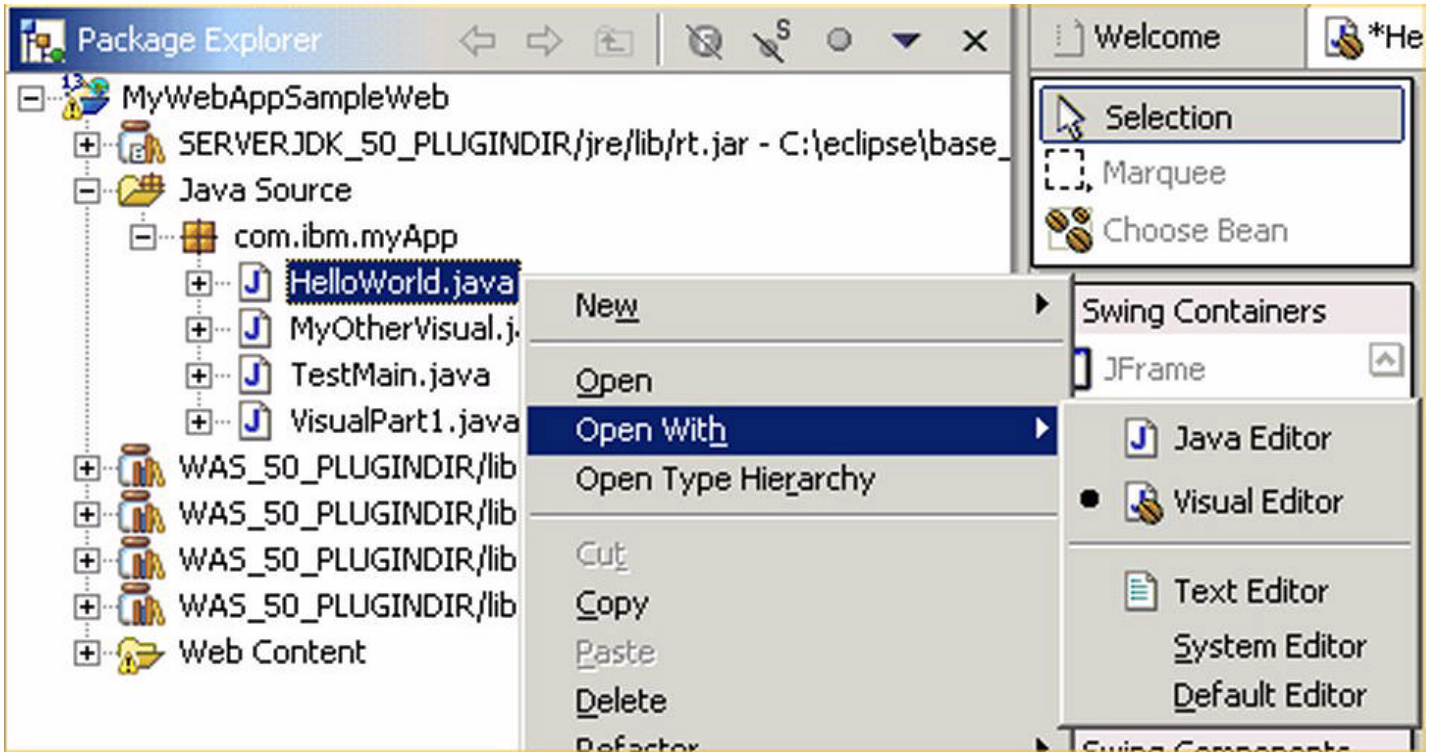
- Learn about Java Visual Editor functions and features
  - Create new Java GUI components
    - Using Swing, AWT
  - Maintain existing GUIs
    - Visually
    - Using the "round-tripping" support
  - Testing GUI components
    - JVM support

What is the Java Visual Editor?

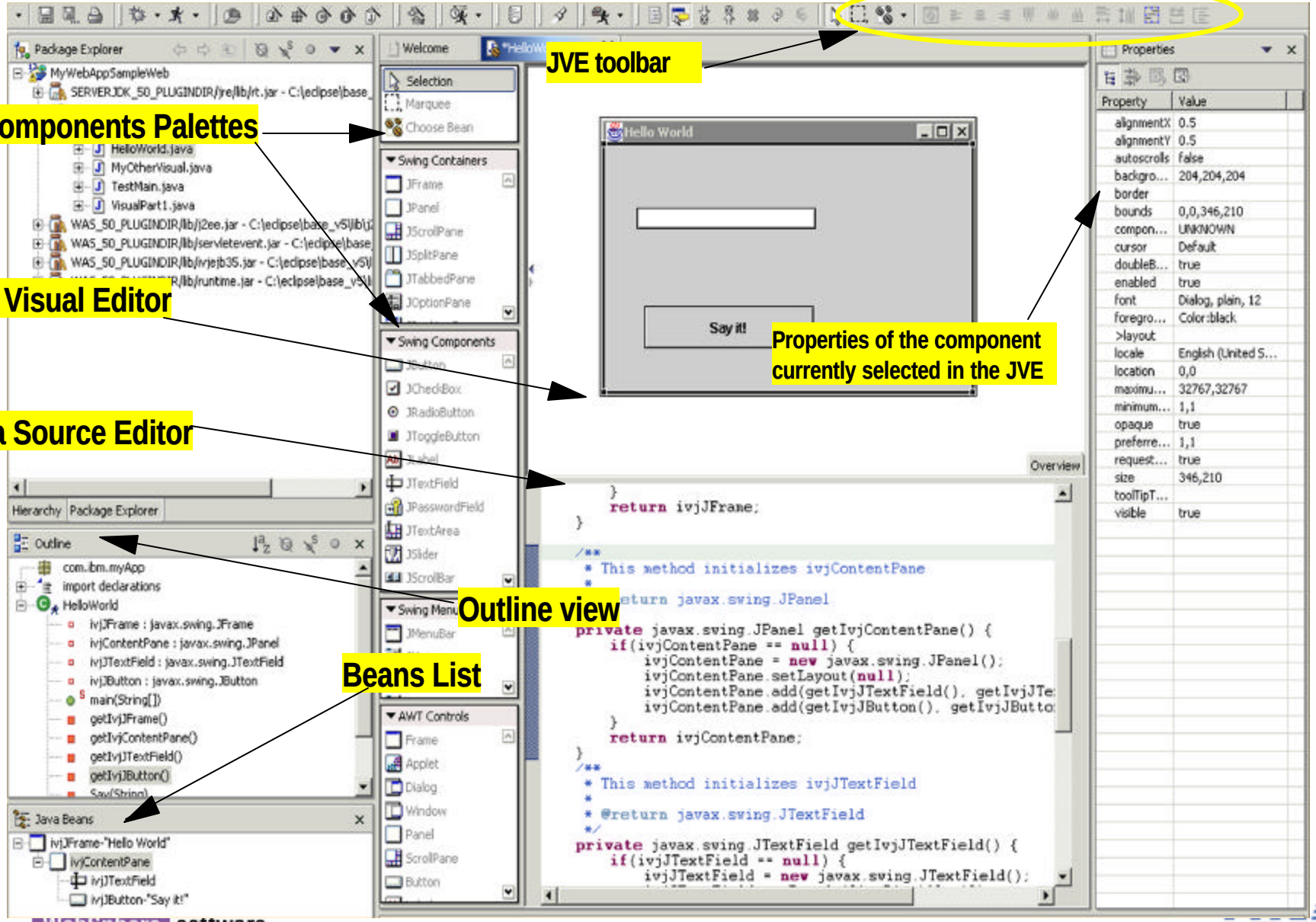
- An editor that allows to view and graphically manipulate GUI parts
  - Primary focus is on the viewing and editing of graphical components
  - A GUI editor, **not** a visual programming environment
- Supports AWT and Swing palettes
  - AWT Components
  - Swing Containers
  - Swing Components
  - Swing Menus
- User-defined beans can also be manipulated with the JVE
- Renders and allows the editing of existing GUI components

Accessing the Java Visual Editor

- A "Java Helper" tool, available from any perspective
- Use "Open with..." on a Java class and select Visual Editor

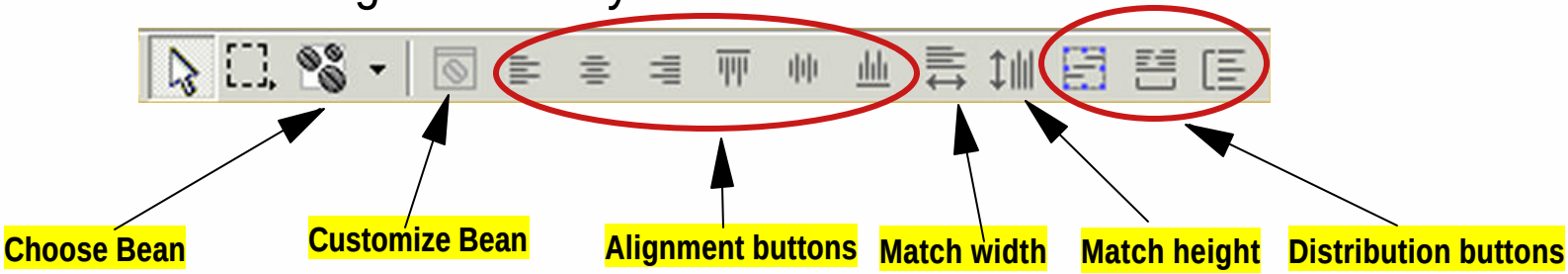


Getting Started with the JVE



Using the Visual Editor

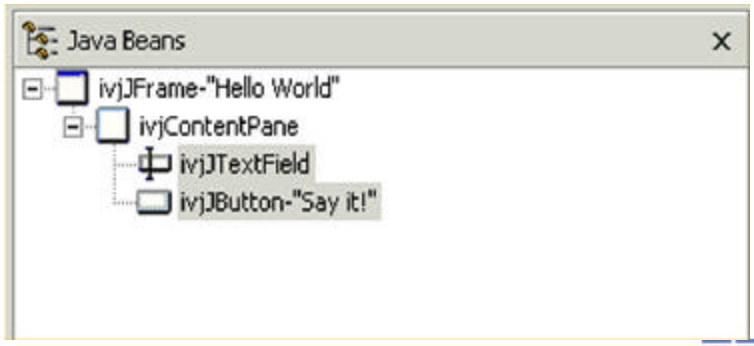
- Create class using Java class wizards and open it using the Visual Editor
- Add GUI beans from palette
  - JPanel's layouts determine size and location of your beans
    - Flow
    - Border
    - Grid
    - GridBag
  - Or, if layout is null, you can position and resize your beans
  - Code gets generated as you drop and modify your beans
- Add any non-GUI bean using the "Choose Bean" function
  - Non-GUI beans become an instance variable in your class
- Use toolbar to align and size your GUI beans



Customizing Beans

- Properties
  - Allows manipulating the BeanInfo data
  - Text editing for String, numbers, and other elemental data
  - Combo editing for Boolean, enumerated lists, etc...
  - Dialog editing for color, font, etc...
  - Nested properties are supported
    - Properties may be JavaBeans themselves

- Beans list
  - Shows children of containers, tables
    - Hierarchical view of the GUI part
  - Allows selecting a bean from the list
    - Useful when you have complex GUIs



JVM Support

- Various JVMs involved
  - One JVM runs WebSphere Studio Site Developer
  - One JVM runs the GUI part
    - Makes it possible to provide an instant rendition of the GUI components and rountripping
  - Support for multiple JREs
- Testing your GUIs
  - If part has *main* method, run as "Java application"
  - If part doesn't have a *main* method, you can still test it
    - Run as "Java Bean"
- Debugging support
  - Debug as a regular Java Application at execution time
  - Or, debug the JVM where the GUI is rendered at development time
    - Use "Debug as..."-> Java Bean option

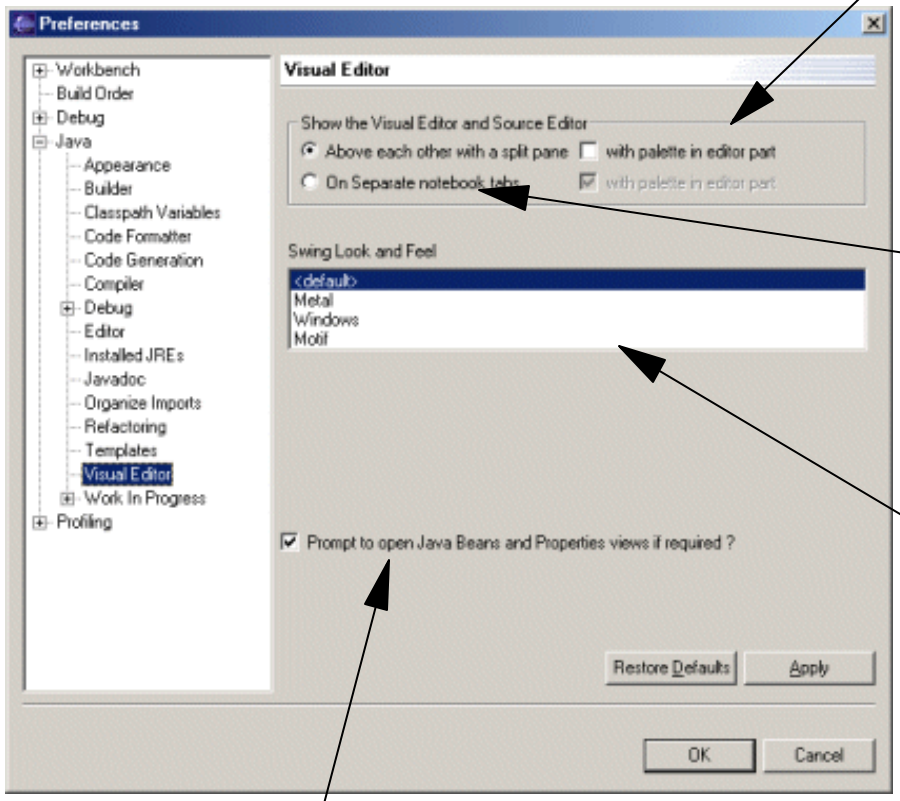
Java Source and Metadata

- Code generation follows certain generation patterns
  - Null constructor
  - Lazy initialization methods for all components
  - No support for event or action listener code generation
    - Manual coding required here
  - No limitations on where user-written code is located
  - Naming convention on variables, methods

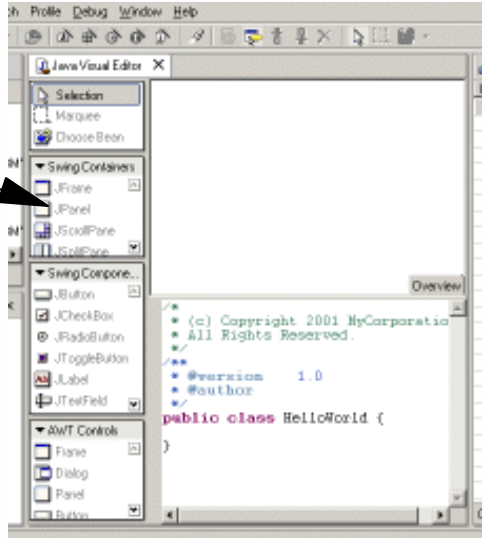
- Metadata
  - Included in the source code as comments
    - All you need is your source code
  - Stores information such as relative position of components on the visual editor, etc.
    - Allows JVE to reconstruct visuals

JavaVisualEditor - Preferences

- Preferences page to configure behavior

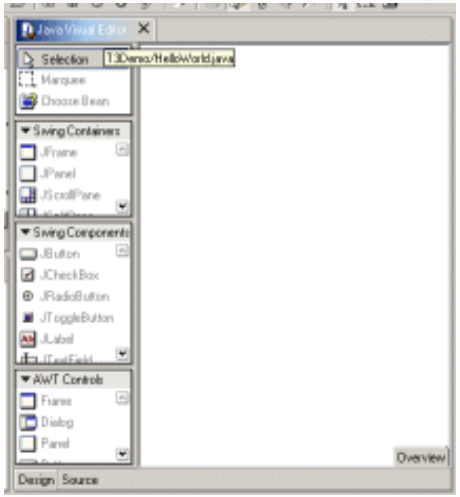


Traditional palette can be used in editor part as well as toolbar JavaBeans



Can show GUI and source in notebook as well as split pane

Swing Look & Feel for editor and for launcher



JavaBeans and Properties viewers automatically opened in Java perspective if required - disabled after first prompt

Step-by-step Guide to Building a GUI with the JVE

- Create a class using the Java class wizard
  - May choose to extend JFrame (not necessary)
  - May or may not have a main method
  - May choose to implement *java.awt.ActionListener* or *java.swing.EventListener*
- Choose beans from palettes and customize GUI appearance
  - Choose containers first (JFrame, JPanel, etc.)
  - Buttons, text fields, list boxes, etc.
  - Customize appearance, position, other characteristics
- Optionally, drop other JavaBeans on the surface
  - Non-visual JavaBeans containing presentation or data manipulation logic
- Manually code action and event listeners

Summary

- Java Visual Editor focuses on the editing of GUI (visual) components
  - Creation, testing, and debugging of new swing-based GUI parts
  - Also allows importing existing GUIs
    - Migration path from VisualAge
- JVE does not aspire to be a full blown visual application builder
  - Manual coding is implied and required
  - JVE greatly facilitates developing GUIs, but doesn't aim at eliminating manual coding
- Round-tripping support
  - Changes to code are rendered in the GUI
  - Changes to the GUI are instantly and incrementally reflected into the source code
- First release
  - More features and functions to come in the future

Section

Appendix

Java Visual Editor versus VAJ Visual Composition Editor (VCE)

- **Two different tools**
  - JVE's focus is on GUI editing and rountripping
  - VCE's ambitions were focused on building applications visually
- **JVE strengths**
  - Supports round-tripping, instant rendition of changes
  - Supports user-defined JREs
  - Smarter Metadata implementation
  - Greater flexibility for user-defined code points in source code
- **JVE limitations in this first release**
  - No connections
  - No support for generating event or action listener logic
  - No support for building applets and no visual support for menus
  - Some parts of the JavaBeans specs are not supported yet
  - No support for non-visual programming constructs (factories, variables, etc.)

Migrating Visual Parts from VAJ to the JVE

- Export the Java code from VisualAge
- Import it in WebSphere Studio Site Developer
  - Open classes that implement visual parts with Visual Editor
  - Rendition in the visual editor should automatically occur
  - The whole VAJ palette is supported
- Migrating Metadata
  - VisualAge Metadata describes relative position of components on the free form surface, connections, and connection bend points
  - Connections and bend points are not supported yet by the Site Developer
    - They may be supported in a future release
  - VisualAge provides a tool to convert the VCE Metadata into comments that are compatible with the way JVE represents it