User Interfaces in XML: The JAXX Framework

Ethan Nicholas

Principal Engineer
Yahoo! Inc.
http://www.jaxxframework.org/

TS-4265

Session Goals

Introducing JAXX

Learn how to build sophisticated user interfaces using JAXX and integrate them into existing Java applications

Agenda

JAXX Basics

Class Tags

Data Binding

Scripting

Styles and Layout

Using JAXX

What Is JAXX?

- JAXX is a Java-based programming language for creating Swing user interfaces
- JAXX is compiled, not interpreted; Each `.jaxx` source file compiles into a Java™ class
- Compiled JAXX classes are ordinary Java classes and can be used just like any other Java class
- Dynamic behaviors are easy with powerful event handling, data binding, and scripting

Why XML? Why JAXX?

Superior Ease of Use

- The Java programming language is fantastic, but it is not especially good at representing user interfaces
- JAXX's XML files are much more compact, while at the same time being easier to read and understand
- XML tags are nested and form a hierarchy—exactly like user interface components
- JAXX was built from the ground up to be a user interface language, and has many features missing in ordinary Java

Agenda

JAXX Basics

Class Tags

Data Binding

Scripting

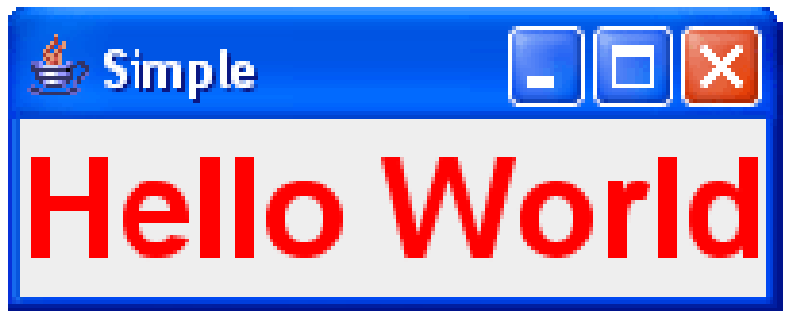
Styles and Layout

Using JAXX

A Simple Example

Simple.jaxx

```
<Application title='Simple'>
  <JLabel text='Hello World' font-size='24'
          foreground='red'/>
</Application>
```



Compiling and running it:

```
> jaxxc Simple.jaxx
> java –cp .;\jaxx\lib\jaxx-runtime.jar Simple
```

Simple.jaxx Examined

Introducing Class Tags

- Most tags in a JAXX file represent instances of Java classes; These are called class tags
- **`<JLabel>`** and **`<Application>`** (a subclass of **`javax.swing.JFrame`**) are class tags
- Most class tag attributes (**`text='Hello World'`**) directly correspond to **`set`** methods
- Putting a component tag inside of a container tag adds the component to the container

Simple.jaxx Examined

Compared to Java Code

JAXX:

```
<JLabel text='Hello World' font-size='24'
            foreground='red'/>
```

Java code:

```
JLabel label = new JLabel();
label.setText("Hello World");
label.setFont(label.getFont().deriveFont(24f));
label.setForeground(Color.red);
```

In general, `foo='value'` is equivalent to `setFoo(value)`

Loading Classes

More on Class Tags

- Class tags can load any Java class
- You can specify fully-qualified class names:

```
<com.mycompany.CustomComponent/>
```

- **`javax.swing`** is imported by default
- Class tags can reference other JAXX files, which will then be compiled (exactly as with Java files)
- JAXX's default support is powerful enough for most components

Using Java Based Expressions

Advanced Attributes

Class tag attributes can be assigned using Java code:

```
<JLabel text='{Math.PI}'
        font='{new Font("Georgia", 0, 24)}'/>
```

This produces: 3.141592653589793

Curly braces allow you to include any Java based expression; This makes it easy to handle unusual data types

Agenda

JAXX Basics

Class Tags

Data Binding

Scripting

Styles and Layout

Using JAXX

A More Complex Example

Sliders.jaxx

```
<Application title='Sliders'>
  <HBox>   <!-- horizontal box -->
    <VBox> <!-- vertical box -- >
      <JSlider id='red' maximum='255'/>
      <JSlider id='green' maximum='255'/>
      <JSlider id='blue' maximum='255'/>
    </VBox>
    <JPanel width='64' height='64'
       background='{new Color(red.getValue(),
                    green.getValue(), blue.getValue())}'/>
  </HBox>
</Application>
```

Sliders Example in Action

What Happens?

Since all three sliders start out at zero, the panel is (perhaps unsurprisingly) black

…but watch what happens when the sliders are adjusted

Data Binding

It Really Is That Simple

- The **`JPanel's`** background attribute was assigned a Java based expression:

```
background='{new Color(red.getValue(),
                       green.getValue(), blue.getValue())}'
```

- JAXX parses the Java based expression and determines its dependencies: **`red.getValue(), green.getValue(),`** and **`blue.getValue()`**
- JAXX knows that, to detect changes in a **`JSlider's getValue()`** method, it must add a **`ChangeListener`** to the slider

Data Binding

What Can JAXX Track?

Data binding can detect changes to the following:

- **`get`** methods corresponding to bound properties
- Fields defined within and modified from JAXX
- Basic **`Component`** and **`Container`** features: **`getX()`**, **`getY()`**, **`getWidth()`**, **`getHeight()`**, **`getComponentCount()`**, etc.
- Anything it was specifically programmed to handle (exhaustive support for Swing)

A full list of bound methods for any given class is available at http://www.jaxxframework.org

Data Binding

More Examples

Automatically enable and disable components:

```
<JButton text='OK'
   enabled='{username.getText().length != 0}'/>
```

Display selected values:

```
<UserPane selectedUser='{users.getSelectedValue()'/>
```

Populate forms:

```
<JTextField text='{currentObject.getId()}'/>
```

Agenda

JAXX Basics

Class Tags

Data Binding

Scripting

Styles and Layout

Using JAXX

Scripting

Java Code Anywhere, Anytime

- Data binding expressions are one form of scripting in JAXX; JAXX uses Java programming language as its scripting language
- Advantages
  - Familiar Java syntax
  - Fully compiled
  - Runs as fast as any other Java code

Scripting

Kinds of Scripts

There are a number of places that Java code may appear in a JAXX file:

- Data binding expressions (escaped with curly braces)
- Event handlers
- Script tags
- Certain special attributes (**`constructorParams`** and **`constraints`**)

Event Handling

No More Anonymous Inner Classes

There is an event handler for each method of each event listener interface; for instance, the **`MouseListener`** interface contains five events:

- **`onMouseClicked`**
- **`onMousePressed`**
- **`onMouseReleased`**
- **`onMouseEntered`**
- **`onMouseExited`**

To create an event listener, assign a snippet of Java code to an event handler attribute:

```
<JButton text='Close' onActionPerformed='dispose()'/>
```

Event Handling

MouseListener.jaxx

```
<JButton label='Mouse Me'
    onMouseClicked= 'demoText.append("Click\n")'
    onMouseEntered= 'demoText.append("Enter\n")'
    onMouseExited=  'demoText.append("Exit\n")'
    onMousePressed= 'demoText.append("Press\n")'
    onMouseReleased='demoText.append("Release\n")'/>
```

Script Tags

Include Arbitrary Java Code

```
<script>
  int count; // fields work with data binding as long as they
             // are defined and manipulated only by JAXX
  private void doClick() {
      count++;
  }
</script>

<JButton label='Clicks: {count}'
          onActionPerformed='doClick()'/>
```

Other Kinds of Scripting

Specify constructor parameters:

```
<script>import java.awt.*;</script>
<BasicStroke id='thickLine' constructorParams='5'/>
```

Provide layout constraints:

```
<app.MainView constraints='BorderLayout.CENTER'/>
```

Agenda

JAXX Basics

Class Tags

Data Binding

Scripting

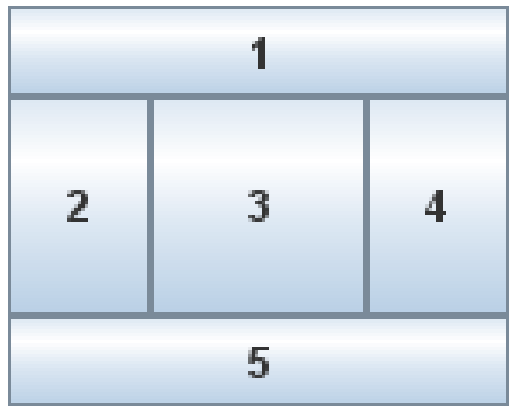
Styles and Layout

Using JAXX

JAXX Layout

Introduction

Between the **`layout`** and **`constraints`** properties, JAXX supports all Java based layout managers




```
<JPanel layout='{new BorderLayout()}'>
  <JButton text='1' constraints='BorderLayout.NORTH'/>
  <JButton text='2' constraints='BorderLayout.WEST'/>
  <JButton text='3' constraints='BorderLayout.CENTER'/>
  <JButton text='4' constraints='BorderLayout.EAST'/>
  <JButton text='5' constraints='BorderLayout.SOUTH'/>
</JPanel>
```

JAXX Layout

Built-in Layouts

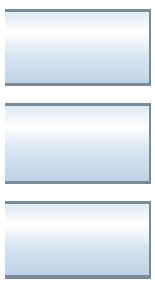
JAXX features three built-in layout components:

<VBox> - Vertical box

<HBox> - Horizontal box

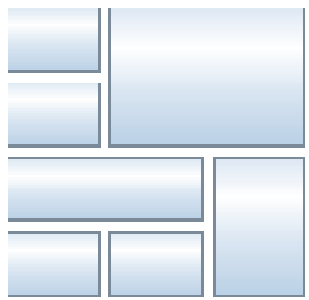
<Table> - Table based on **GridBagLayout**

VBox:



HBox:

Table:

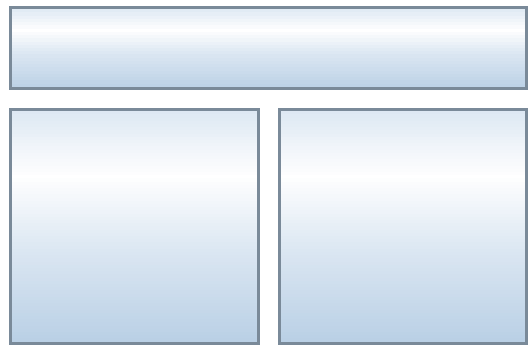


Tables

Like GridBagLayout, but Easy to Use

```
<Table fill='both' insets='2, 2, 2, 2'>
  <row>
    <cell columns='2'><JButton/></cell>
  </row>

  <row weightx='1' weighty='1'>
    <cell><JButton/></cell>
    <cell><JButton/></cell>
  </row>
</Table>
```



Simple Table

CSS Stylesheets

Separate Style from Content

JAXX files can include CSS stylesheets:

```
<style>
  JButton.fancy {
    foreground: #30cc60;
    font-size: 24;
  }
</style>

<JButton label='Styled Button' styleClass='fancy'/>
```

CSS Selectors

Choosing Components to Style

JAXX supports four different kinds of CSS selectors:

`JButton` - **`JButton`** and its subclasses

`#ok` - Objects with **`id='ok'`**

`.fancy` - Objects with **`styleClass='fancy'`**

`:enabled` - Components which are currently enabled; This is one example of a pseudoclass

CSS Pseudoclasses

Easy Dynamic Effects

Because pseudoclasses depend upon the runtime state of a component, they are inherently dynamic

```
<style>
  AbstractButton:mouseover {
    foreground: red;
  }
</style>
```

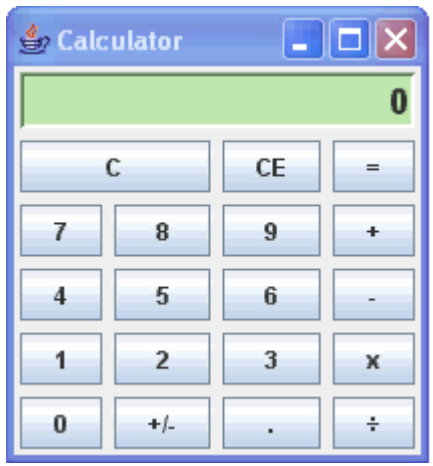
Supported pseudoclasses include **`mouseover`**, **`mouseout`**, **`focused`**, **`unfocused`**, **`enabled`**, **`disabled`**, **`armed`**, **`unarmed`**, **`selected`**, and **`unselected`**

Programmatic Pseudoclasses

Ultimate Flexibility

A programmatic pseudoclass is a Java based expression that decides whether or not a particular component should be styled

```
<style>
  #display:{object.getText().startsWith("-")} {
    foreground: red;
  }
</style>
```



Agenda

JAXX Basics

Class Tags

Data Binding

Scripting

Styles and Layout

Using JAXX

JAXX Requirements

The Most Important Rule? Have Fun!

JAXX is licensed under the BSD License; it is free for both commercial and non-commercial use, and may be freely modified

The JAXX compiler requires the Java Development Kit version 1.4.0 or higher. Compiled objects require the Java Runtime Environment 1.4.0 or higher

`jaxx-runtime.jar` (about 30K as of this writing) must be on the class path

Integrating with Java

From Java to JAXX

Compiled JAXX classes are ordinary Java classes; Java code can instantiate and manipulate them the same as any other class

All JAXX classes have a method **`getObjectById()`** which allows access from Java code; You can also define your own methods, fields, and constructors to expose whatever functionality you need

Integrating with Java

From JAXX to Java

JAXX's class tags can create and configure any Java class:

```
<com.mycompany.MyComponent widgets='26'/>
<!-- you don't have to register the component! -->
```

If that is not enough, you can use **<script>** tags to do…well, anything

JAXX integrates with Java more easily than any other XML user interface language

Summary

- JAXX is much easier and faster than handcoding Swing GUIs
- It is (to my knowledge) the only compiled user interface language for Java technology
- JAXX produces ordinary Java classes, and…
- It has powerful features you won't find anywhere else

For More Information

The Official Web Site

http://www.jaxxframework.org

- Source and binary distributions
- Complete documentation
- Web Start-able demos
- Community forums

Q&A

Ethan Nicholas

http://www.jaxxframework.org/

User Interfaces in XML: The JAXX Framework

Ethan Nicholas

Principal Engineer
Yahoo! Inc.
http://www.jaxxframework.org/

TS-4265