Apache Royale
Tour De Jewel

01.10.2020 18:55 CEST

Carlos Rovira & Andrew Wetmore
Journey

Jewel
Creating and building the starter Apache Royale Jewel Crux Project.

Components
Creating the initial application visuals.

Layouts
How Jewel Components display in the UI

Themes
The look & feel of Jewel
Jewel

Know what is Jewel, the UI Set behind Tour de Jewel
Welcome to Apache Royale Tour de Jewel Component Explorer v.0.9.7

Choose your Jewel Theme:
- Primary: Blue
- Secondary: Teal
- Emphasis: Emerald
- Flat: 

Jewel is a themeable and responsive set of user interface components for Apache Royale to help you quickly build the front end of your applications with ActionScript & MXML.

Check https://royale.apache.org

Jewel Browser support: Chrome 44+, Firefox 31+, IE 11, Edge 15+, Safari 9.1.1, Opera 12.1+, Android 4.0.3, Windows, MacOS 10.11, Linux.
Jewel ComboBox

Jewel ComboBox is a component that displays an input field with a button to show and pop-up list with selections. Selecting an item from the pop-up list places that item into the input field.

String Collection

This ComboBox is using a simple string collection as dataProvider and a ComboBoxTextPrompt bead to show a prompt message.

Object Collection

This ComboBox is using an object collection as dataProvider. Use labelField to indicate the object property to use as label. A ComboBoxTextPrompt bead is used to show a prompt message.

Selected Index

String value: Doctor Manhattan

- Avengers Team...
- Iron Man
- Hulk
- Thor
- Captain America
- Black Widow

Jewel List

Jewel List examples.

Default

- Blueberries
- Bananas
- Oranges
- Lemons
- Watermelons
- Apples

ItemRenderer and rowHeight=52

Notice variableRowHeight should be false to use rowHeight. Click on render's icon will remove that renderer.
Why Royale?

First why choose Royale instead other mainstream tech?

- **Supported by Apache.** No Big Corporations driving decisions for its interest.
- **Best of Flex and web dev.** Brings the best of latest 20 years in web development.
- **Multi-Target.** Right now: JS, SWF. In the future: WASM? iOS? Android?.
- **Compiled / Transpiled Apps.** Processing of source code
- **Powerful features (not available any other technology):**
  - **MXML.** Declarative language is the real power behind Royale.
  - **AMF protocol.** Best performance for RPC calls (over JSON). **RemoteObject** to interact with Java, .NET, PHP, Python,...
  - **Crux + Metadata.** IOC (Inversion of Control), DI (Dependency Injection).
  - **Data Binding.** Change model values and update views and use in conjunction with MXML
  - **Modules.** Separate your app in pieces and load independently.
  - **Performance:** Is amazing if we compared with other current mainstream tech (i.e React)
    - (*) See comparison image for the same component Royale vs React.
  - **OOP, Strongly typing, Error Checking...**
  - **Composition:** PAYG & Strand & Beads.
Royale UI Sets

- Different UI Sets
  - Basic
  - Express
  - Jewel
  - Emulation (for Flex MX/Spark)

So...
Why Jewel?

- Modern browser web development
  - Modern components (i.e: Drawer, Grid, Card...)
  - Support Browser Responsiveness
- Style customization
  - Themeable (libraries to change all UI component styles at once)
  - Set of predefined themes (144 different combinations)
- Components adapt to different screens (i.e: DateField)
Jewel Considerations

- Get the best of what we had with Flex and modern web development
  - **Best of Flex** (i.e: MXML, RemoteObject and AMF,...)
  - **Best of Browsers** (i.e: CSS, No plugins,...)
- **PAYG**
  - But with a bit of balance to keep in mind real world needs (Jewel is a bit more permissive than Basic).
- **Styling**
  - Promote **SASS** over CSS (but don’t enforce for end users)
### Browser Support

<table>
<thead>
<tr>
<th>Browser</th>
<th>Minimum Version</th>
<th>Release Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desktop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google Chrome</td>
<td>44+</td>
<td>July 21, 2015</td>
</tr>
<tr>
<td>Mozilla Firefox</td>
<td>34+</td>
<td>Dec 1, 2014</td>
</tr>
<tr>
<td>Apple Safari</td>
<td>11.1+ (High Sierra)</td>
<td>Feb 22, 2018</td>
</tr>
<tr>
<td>Microsoft Edge</td>
<td>15+</td>
<td></td>
</tr>
<tr>
<td><strong>Microsoft IE</strong></td>
<td><strong>11+</strong></td>
<td></td>
</tr>
<tr>
<td>Opera</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mobile</td>
<td></td>
<td></td>
</tr>
<tr>
<td>iOS SafariMobile</td>
<td>11.0+</td>
<td>Sep 19, 2017</td>
</tr>
<tr>
<td>Android</td>
<td>5.0</td>
<td>2014</td>
</tr>
<tr>
<td>Windows Mobile</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

https://apache.github.io/royale-docs/component-sets/jewel#browser-support
Basic vs Jewel

- Basic is:
  - Foundation of Royale components and beads.
  - 100% PAYG.
  - The most basic representation of components.
    - No additional structure considerations for visuals.
    - Visuals are very raw (no fonts, margins, colors,...mostly black/white)
  - SWF and JS considered.

- Jewel is:
  - Build on top of Basic Foundation.
  - PAYG (but not 100%).
  - Advanced representation of components.
    - Internal structures to build good visuals.
    - Separation of styles between structure (jewel.swc) and styling (themes.swc).
  - Only JS considered (for now, but could be extended in the future).
Components

Jewel set of available components and containers ready to use
Components

● Containers and Components
  ○ Containers can hold an arbitrary set of other containers and/or components. Can be populated via code or data.
  ○ Components has a concrete functionality
  ○ 60+ containers and components and growing.
  ○ For a full list see:
    ■ https://apache.github.io/royale-docs/component-sets/jewel#components

● Beads
  ○ Common (i.e: Disabled) and Specialized (PasswordInput for TextInput)
**Containers (Code Based Population)**

- View / ResponsiveView
- Group / Container
  - HGroup, VGroup / HContainer, VContainer
- Card
- Grid & GridCell
- SectionContent / ScrollableSectionContent / TabBarContent
- Form / FormItem
- Drawer
- Module
- Others (Wizard, TopAppBar, ...)
Containers (Code Based Population)

```xml
<j:Container width="100%">
  <j:beads>
    <j:TileHorizontalLayout localId="th1" verticalGap="6" horizontalGap="6" requestedColumnCount="3"/>
  </j:beads>

  <html:Div className="box" text="1"/>
  <html:Div className="box" text="2"/>
  <html:Div className="box" text="3"/>
  <html:Div className="box" text="4"/>
  <html:Div className="box" text="5"/>
  <html:Div className="box" text="6"/>
  <html:Div className="box" text="7"/>
  <html:Div className="box" text="8"/>
  <html:Div className="box" text="9"/>
  <html:Div className="box" text="10"/>
  <html:Div className="box" text="11"/>
  <html:Div className="box" text="12"/>

  </j:Container>
```
Containers (Data Based Population)

- DataContainer
- List
  - TabBar
  - ButtonBar, IconButtonBar, ...
  - Navigator
- ComboBox
- DropDownList
- Table, SimpleTable
- DataGrid

Considerations:

- Data populate through dataProvider.
  - ArrayList
  - ArrayListView (for sort and filtering)
- ItemRenderer (data item visualization)
Containers (Data Based Population)
Components

- Button, IconButton, ToggleButton
- TextInput, IconTextInput, TextArea
- CheckBox, RadioButton
- DateField, DateChooser
- Image, Icon
- Label
- NumericStepper
- Slider
- Others (PopUp, SimpleLoader,...)

- Based on **StyledUIBase** (that extends **Basic UIBase**)
- Composed by Beads (Model, View, Controller, Layout, ...)
- Beads are configured in CSS
Components

```xml
<j:Card visible="true" visible.loggedIn="false">
    <html:H1 text="View States with dot notation"/>
    <j:TextInput text="someuser"/>
    <j:TextInput text="somepass">
        <j:beads>
            <j:PasswordInput/>
        </j:beads>
    </j:TextInput>
    <j:Button text="Login" emphasis="primary" click="currentState = 'loggedIn'"/>
</j:Card>
```
Beads

- **Common**
  - Disabled
  - SizeControl
  - ToolTip
  - ResponsivSize

- **Specialized**
  - TabBar -> AssignTabContent
  - TexInput and TextArea -> TextPrompt
Default Beads

- Defined through CSS (SASS):
- i.e: List is composed of the following beads:

```java
j|List
IBeadView: ClassReference("org.apache.royale.jewel.beads.views.ListView")
IBeadController: ClassReference("org.apache.royale.jewel.beads.controllers.ListSingleSelectionMouseController")
IBeadKeyController: ClassReference("org.apache.royale.jewel.beads.controllers.ListKeyDownController")
IItemRendererClassFactory: ClassReference("org.apache.royale.core.SelectableItemRendererClassFactory")
IItemRenderer: ClassReference("org.apache.royale.jewel.itemRendererers.ListItemRenderer")
IItemRendererInitializer: ClassReference("org.apache.royale.jewel.beads.itemRendererers.ListItemRendererInitializer")
ISelectableItemRenderer: ClassReference("org.apache.royale.jewel.beads.itemRendererers.ClassSelectoListSelectableItemRendererBead")
IViewport: ClassReference("org.apache.royale.jewel.supportClasses.scrollbar.ScrollingViewport")
IBeadModel: ClassReference("org.apache.royale.jewel.beads.models.ArrayListSelectionModel")
IDataProviderItemRendererMapper: ClassReference("org.apache.royale.html.beads.SelectionDataItemRendererFactoryForCollectionView")
```
How Jewel components are displayed in screen
Layout Beads

- Based on CSS “flex”
- Tries to delegate layout to the browser when possible
- BasicLayout is for absolute positioning
- HorizontalLayout / VerticalLayout
- HorizontalCenteredLayout / VerticalCenteredLayout
- much more...
Layouts Beads

- TileHorizontalLayout uses a combination of css flex and AS3
Responsive Layout

- Grid / GridCell
  - GridLayout / GridCellLayout
- Sizes can adapt to screen width
  - Fonts size change between mobile / tablet / desktop
- ResponsiveView (instead of View)
- Beads:
  - ResponsiveVisibility
  - ResponsiveSize
  - ...
The look & feel of Jewel
Themes
About themes

- **JewelTheme** is the master theme
- Use **SASS**
- Light/Dark and Flat/No Flat variations
- 12 Colors
- primary, secondary and emphasized
- 144 variations
- We can use
  - Jewel Theme (if you can compile SASS)
  - A combination of three themes (primary, secondary and emphasized)
- Dark themes still WIP
Themes
Runtime
Switching
Jewel Links

- Tour de Jewel
  - https://royale.apache.org/tourdejewel
- Jewel Blog Examples
  - https://royale.apache.org/category/jewel-ui-set/
- Jewel Documentation
  - https://apache.github.io/royale-docs/component-sets/jewel
Future

A look into the Jewel future
Jewel's Future

- Continue improving the UI Set
  - Continue improving actual components
    - Provide missing functionality
    - Bug fixing
  - Develop new features
    - Components still missing
    - new beads
  - Dark Themes
Where to go from here

- Join our mailing list (“users” or “dev”)
  - [https://royale.apache.org/mailing-lists/](https://royale.apache.org/mailing-lists/)

- Do you want to join us?
  - [https://royale.apache.org/get-involved/](https://royale.apache.org/get-involved/)

- StackOverFlow
  - [https://stackoverflow.com/questions/tagged/apache-royale](https://stackoverflow.com/questions/tagged/apache-royale)

- If you care this project -> Follow us on Social Networks!! :) 
  - **Twitter:** [https://twitter.com/apacheroysale](https://twitter.com/apacheroysale)
  - **LinkedIn:** [https://www.linkedin.com/groups/12118437](https://www.linkedin.com/groups/12118437)
  - **FaceBook:** [https://www.facebook.com/ApacheRoyaleSDK](https://www.facebook.com/ApacheRoyaleSDK)
Thank you for your attention!

Questions?

royale.apache.org