Your presenters

- Alina Kazi - Apache Royale committer; headed the migration project we discuss here.

- Andrew Wetmore - Apache Royale committer working on documentation; editor/writer for the Apache Infrastructure team
The challenge

**DBIZ Solutions** had a mission-critical Flex app of over 1456 .MXML files, 666 .AS files. Required the Flash Player or AIR (desktop Flash environment) to run.

With the end of Flash support they had to:

- rebuild the whole app in a new technology

  OR

- move the app to something that would support their existing assets and team skills
The decision & Alina's journey

DBIZ Solutions chose to migrate from Adobe Flex to Apache Royale.

- Flex and Royale are very similar, requiring similar team skills
- For the differences in Royale, learning curve is gentle
- Both use .mxml and .as, so no need to learn a new code language
- Shorter time to develop the replacement app, with fewer risks
The steps

The migration took these steps

- API report
- The easy parts
- Create stubs
- Emulation
- Component set
- Compile - run - find bugs - fix - repeat
- Success!
API report

● What components does the app use?
● Which map to existing Royale components?
● Which need emulation?

The easy parts

- Most ActionScript code from Flex "just runs" in Royale
- Simple to change namespaces
  
  Flex: `xmlns:s="library:/ns.adobe.com/flex/spark"` becomes
  Royale: `xmlns:s="library://ns.apache.org/royale/spark"

- Replace `<mx:Script>` to `<fx:Script>`
- Replace `<mx:Component>` to `<fx:Component>`
- `<mx:beads>` `<js:ClassAliasBead />` `<mx:beads>`

Add this to the application file, and in every module file add basic Namespace at the top `xmlns:js="library://ns.apache.org/royale/basic"`
Less easy

Moving from Flash to Royale equivalents, or emulations:

- **Flex:** flash.net.URLLoader  becomes  
  Royale: org.apache.royale.net.URLLoader

- **Flex:** flash.core.UIComponent  becomes  
  Royale: mx.core.UIComponent

- **Flex:** flash.display.DisplayObject  becomes  
  Royale: mx.core.UIComponent or mx.core.IUIComponent

- **Flex:** flash.events.Event  becomes  
  Royale: org.apache.royale.events.Event
Less easy

- Flex: `flash.net.navigateToURL` becomes Royale: `org.apache.royale.core.BrowserWindow`
- Flex: `flash.utils.getQualifiedClassName` becomes Royale: `org.apache.royale.reflection.getQualifiedClassName`

These are just a few examples,

Find what you require to make your Application run (see in the API Report you generated) and find the equivalents in the Apache Royale SDK.
Create stubs

Create Royale components and functions that connect with the app but may not actually do everything the Flex component did, to let the app compile.

Then replace each stub with code that does what the app needs.

Flex SDK: https://github.com/apache/flex-sdk

Royale SDK: https://github.com/apache/royale-asjs
Emulation

When the app needs a component Royale does not yet have, create an equivalent component:

- remove dependencies on Flash
- follow Royale patterns such as strands and beads so the component can do what the app needs
- Not necessarily backward compatible with the existing Flex app

https://github.com/apache/royale-asjs/wiki/emulation-components
Component sets

- Flex components have bulky code to cover all sorts of possible uses
- Royale follows the **PAYG** principle
  - Strands and beads
- Existing Royale component sets range from very lightweight for quick demos to richer component sets like Jewel
- This project opted to emulate 80-90% of the components, and only used strands and beads in one area
Compile - run - find bugs - fix - repeat

- Patient, hard work by two developers
- Strong support and participation by the Royale community
- Found and fixed hundreds of bugs...
Bug:
Problem with the dropdown menu

| A   | B               | C       | D                             | E       | F   | G   | H   | I   | J   | K   | L   |
|-----|----------------|---------|--------------------------------|---------|-----|-----|-----|-----|-----|-----|-----|-----|
| 6   | TS/158/BOI/001 | All     | System should not allow users to continue changes in entries without pressing OK or cancel button on warning/validation/confirmation messages. | AC      | 6   | O   | 2   |     |     |     |     |
| 15  | TS/158/BOI/001 | All     | Buttons should be designed as clickable buttons like in flex system. | AC      | 6   | O   | 2   |     |     |     |     |
| 8   | TS/158/BOI/001 | All     | System blanks out on saving and deletion of entries. | AC      | 6   | O   | 2   |     |     |     |     |
| 9   | TS/158/BOI/001 | All     | There is a blank left grey when there is no entry system should show rows with or without data. | AC      | 6   | O   | 2   |     |     |     |     |
| 10  | TS/158/BOI/001 | All     | There should be a column selector. | AC      | 6   | O   | 2   |     |     |     |     |
| 11  | TS/158/BOI/001 | All     | Version method should be same as password. | AC      | 6   | O   | 2   |     |     |     |     |
| 12  | TS/158/BOI/001 | All     | Enter is not working for login. | AC      | 6   | O   | 2   |     |     |     |     |
| 13  | TS/158/BOI/001 | All     | There should be drop-down arrow for sub-menu instead of question mark. | AC      | 6   | O   | 2   |     |     |     |     |
| 14  | TS/158/BOI/001 | All     | There should be drop-down icon for sorting file in flex. | AC      | 6   | O   | 2   |     |     |     |     |
| 15  | TS/158/BOI/001 | All     | If application is taking time to load page then there should be some indication for users to wait like loading clock in flex applications. | AC      | 6   | O   | 2   |     |     |     |     |
| 24  | TS/158/BOI/001 | All     | Drop-down should be same as flex application. | AC      | 6   | O   | 2   |     |     |     |     |
| 17  | TS/158/BOI/001 | All     | On clicking save button system should highlight all mandatory fields if user has not provided data in them. | FZ      | 6   | O   | 2   |     |     |     |     |
| 19  | TS/158/BOI/001 | All     | Display code field disabled as in flex after saving. | FZ      | 6   | O   | 2   |     |     |     |     |
| 27  | TS/158/BOI/001 | All     | Fix page resolution according to screen resolution. | FZ      | 6   | O   | 2   |     |     |     |     |
| 30  | TS/158/BOI/001 | All     | Tab should shift cursor to next verifiable field. | FZ      | 6   | O   | 2   |     |     |     |     |
Bug:

Need highlighting for the text entry field
Bug:

When you open a new submenu, any submenu that is already open should close.
Success!

- Royale version compiles and runs reliably and accurately.
- The original Flex app required ten full-time developers working for 13 years. Porting to Royale took two developers (plus community help) just two years.
- The company, its staff, and its end users are happy
Success!
Success!
Very thankful to

- Mr. Ahsan Akhtar, Team Manager at DBIZ Solutions, who provided the opportunity.
- Alex Harui, PMC Member of Apache Royale, who helped a lot with the Emulation Components.
- The rest of the Royale developer community who devoted many hours to making the migration a success.
- My aunt Razia, my father, my mother, my husband, my kids, my sisters and my brothers
Thank you for your attention!

Questions?

royale.apache.org