Open Iteration Meeting

Vera Carr
08.18.09
Context

- Flex 4 introduces a new component set (Spark) and maintains an existing set (MX / Halo)
- Spark in Flex 4 has a subset of MX / Halo components (goal is to have parity in Flex 5)
- Need usability improvements to ensure success with Flex 4
Agenda

- Challenges
- Resolutions
- Positioning
  - Additional Features
- Roadmap
  - Community Features
Agenda

- Challenges
- Resolutions
- Positioning
- Additional Features
- Roadmap
- Community Features
Challenges

- Confusion using Spark and MX together
- Some Spark components don't work in MX containers
- Some Spark components don’t have 1:1 functionality with MX
- Fewer styling capabilities than previous version
- Hard to mix FTE and TextField
- Lack of documentation; cutting and pasting code will sometimes fail
Agenda

- Challenges
- **Resolutions**
- Positioning
- Additional Features
- Roadmap
- Community Features
Confusion using Spark and MX together

- Addressed through Builder and documentation
- Refer to Halo components as MX
- Builder select preferred component set: MX only or MX/Spark
- Builder code hinting will recommend most appropriate component
Some Spark components don't work in MX containers

Resolution

- Support MX Navigators With Spark Containers
- Spark Form

Not Resolved

- Spark Image
Some Spark components don't work in MX containers

Resolution

- Support MX Navigators With Spark Containers
  - Spark Form
  - Spark Image

Not resolved

Dev: Alex / QA: Rob

Details:

- Allow use of MX Container and Spark SkinnableContainer in MX ViewStack, TabNavigator, and Accordian
- Group will not be supported
Some Spark components don't work in MX containers

Resolution

- Support MX Navigators With Spark Containers

  - Spark Form

Not resolved

- Spark Image

Dev: Ryan / QA: Jacob

Details:

- Spark equivalent of MX Form
Some Spark components don't work in MX containers

Resolution
- Support MX Navigators With Spark Containers
- Spark Form

Not Resolved
- Spark Image

Problem
- MX containers don't support GraphicElement primitives such as BitmapImage primitive – needs to be a UIComponent

Work around
- Except in optimized instances use MX Image
Some Spark components don’t have 1:1 functionality with MX

Resolution

- ControlBar in Panel and Application
- Editable ComboBox
- List Drag and Drop
- Paged Data in Spark List
- Spark ButtonBar support for MX Viewstack
- TabBar
- TileGroup
- TitleWindow (PopUpPanel)

Not resolved

- Spark Divider
- IconButton
Some Spark components don’t have 1:1 functionality with MX

Resolution

- **ControlBar in Panel and Application**
  - Editable ComboBox
  - List Drag and Drop
  - Paged Data in Spark List
  - Spark ButtonBar support for MX Viewstack
  - TabBar
  - TileGroup
  - TitleWindow (PopUpPanel)

Dev: Alex / QA: Steven

Details:

- Provide a “ControlBar” for Spark Panel and Application, similar to the ControlBar for MX Panel
- ControlBarLayout (subclass of BasicLayout) is the assigned layout for PanelSkin and ApplicationSkin

Not resolved

- Spark Divider
- IconButton
Some Spark components don’t have 1:1 functionality with MX

Resolution

- ControlBar in Panel and Application
- **Editable ComboBox**
  - List Drag and Drop
  - Paged Data in Spark List
  - Spark ButtonBar support for MX Viewstack
  - TabBar
  - TileGroup
  - TitleWindow (PopUpPanel)

Not resolved

- Spark Divider
- IconButton

Dev: Jason / QA: Jacob

Details:

- ComboBox component is a DropDownList with a TextInput control instead of a label
- User can type into the TextInput control to filter out the set of possible values
Some Spark components don’t have 1:1 functionality with MX

Resolution

- ControlBar in Panel and Application
- Editable ComboBox
- **List Drag and Drop**
  - Paged Data in Spark List
  - Spark ButtonBar support for MX Viewstack
- TabBar
- TileGroup
- TitleWindow (PopUpPanel)

Not resolved

- Spark Divider
- IconButton

Dev: Evtim / QA: Ella

Details:

- Provide drag and drop support for List on par with the support offered by the MX List component
Some Spark components don't have 1:1 functionality with MX

Resolution

- ControlBar in Panel and Application
- Editable ComboBox
- List Drag and Drop
- Paged Data in Spark List
- Spark ButtonBar support for MX Viewstack
- TabBar
- TileGroup
- TitleWindow (PopUpPanel)

Not resolved

- Spark Divider
- IconButton

Dev: Hans / QA: Rob

Details

- Eliminating the need to handle ItemPendingErrors thrown by a DataGroup's dataProvider for list with remote paged data
Some Spark components don’t have 1:1 functionality with MX

Resolution

- ControlBar in Panel and Application
- Editable ComboBox
- List Drag and Drop
- Paged Data in Spark
- **Spark ButtonBar support for MX Viewstack**
- TabBar
- TileGroup
- TitleWindow (PopupPanel)

Not resolved

- Spark Divider
- IconButton

Dev: Alex / QA: Rob

Details

- Allow Spark ButtonBar to take an MX ViewStack as a dataProvider
- ButtonBar dataProviders must implement IList, so ViewStack gets modified to implement IList on its list of children
Some Spark components don't have 1:1 functionality with MX

Resolution

- ControlBar in Panel and Application
- Editable ComboBox
- List Drag and Drop
- Paged Data in Spark List
- Spark ButtonBar support for MX Viewstack

- TabBar
  - TileGroup
  - TitleWindow (PopUpPanel)

Not resolved

- Spark Divider
- IconButton

Dev: Glenn / QA: Joann

Details:

- TabBar is a new SkinnableComponent that is a subclass of ButtonBar
Some Spark components don’t have 1:1 functionality with MX

Resolution

- ControlBar in Panel and Application
- Editable ComboBox
- List Drag and Drop
- Paged Data in Spark List
- Spark ButtonBar support for MX Viewstack
- TabBar
- TileGroup
- TitleWindow (PopUpPanel)

Not resolved

- Spark Divider
- IconButton

Dev: Steve / QA: Steve

Details

- Provide convenience class for creating a Group with TileLayout
- VGroup and HGroup are already provided, need TileGroup for consistency
Some Spark components don’t have 1:1 functionality with MX

Resolution
- ControlBar in Panel and Application
- Editable ComboBox
- List Drag and Drop
- Paged Data in Spark List
- Spark ButtonBar support for MX Viewstack
- TabBar
- TileGroup

- TitleWindow (PopUpPanel)

Not resolved
- Spark Divider
- IconButton

Dev: Kevin & Glenn / QA: Jody

Details
- Create a Spark TitleWindow equivalent to MX TitleWindow
Some Spark components don’t have 1:1 functionality with MX

Resolution

- ControlBar in Panel and Application
- Editable ComboBox
- List Drag and Drop
- Paged Data in Spark List
- Spark ButtonBar support for MX Viewstack
- TabBar
- TileGroup
- TitleWindow (PopUpPanel)

Not resolved

- Spark Divider
- IconButton

Problem

- No drag-able divider control that can be placed in Spark containers

Work around

- Use MX DividedBox – it can take any UIComponent
Some Spark components don’t have 1:1 functionality with MX

Resolution
- ControlBar in Panel and Application
- Editable ComboBox
- List Drag and Drop
- Paged Data in Spark List
- Spark ButtonBar support for MX Viewstack
- TabBar
- TileGroup
- TitleWindow (PopUpPanel)

Not resolved
- Spark Divider
- IconButton

Problem
- Spark has no built-in support for adding an icon to a button

Workaround
- Create a skin to display the icon
- Extend Spark Button to handle icons
Fewer styling capabilities than previous version

Resolution

- More Styles for Spark Skins
- BorderContainer
- Compiler Errors for Other Theme's Styles
- Per-Module Style Management

Not resolved

- Slider Tick Marks
Fewer styling capabilities than previous version

Resolution

- More Styles for Spark Skins
  - BorderContainer
  - Compiler Errors for Other Theme's Styles
  - Per-Module Style Management

Not resolved

- Slider Tick Marks

Dev: Glenn / QA: Joann

Details:

- Adding new style properties for the Spark skins
- Expose the most commonly requested styles
Fewer styling capabilities than previous version

Resolution

- More Styles for Spark Skins
- **BorderContainer**
- Compiler Errors for Other Theme's Styles
- Per-Module Style Management

Not resolved

- Slider Tick Marks

Dev: Jason / QA: Jacob

Details:

- A container class that can be styled with a border and a background fill or image
Fewer styling capabilities than previous version

Resolution
- More Styles for Spark Skins
- BorderContainer
- Compiler Errors for Other Theme's Styles
- Per-Module Style Management

Not resolved
- Slider Tick Marks

Dev: Paul / QA: Gaurav

Details
- Validate MXML style attributes for styles that are in our known themes (Halo and Spark)
Fewer styling capabilities than previous version

Resolution

- More Styles for Spark Skins
- BorderContainer
- Compiler Errors for Other Theme's Styles
- Per-Module Style Management

Not resolved

- Slider Tick Marks

Dev: Darrell / QA: Gaurav

Details:

- Allow styles to be localized to a particular module or sub-application
- Styles declared in module will not be visible by sibling modules or sub-applications
Fewer styling capabilities than previous version

Resolution
- More Styles for Spark Skins
- BorderContainer
- Compiler Errors for Other Theme's Styles
- Per-Module Style Management

Not resolved
- Slider Tick Marks

Problem
- MX Sliders had built in support for tick marks

Work around
- Implement a custom skin
Hard to mix FTE and TextField

Resolution

- TLFTextField support for MX components

Dev: Gordon / QE: Peter

Details:

- Ability for MX components to use FTE/TLF-based text instead of TextField-based text
- Certain MX components with Spark equivalents will need to port over to using the Spark component
Lack of documentation

- Flood Flex Dev Center and blogs with Flex 4 samples
- Update Tour de Flex with Flex 4 samples
Agenda

- Challenges
- Resolutions
- **Positioning**
  - Additional Features
- Roadmap
- Community Features
Positioning

- Flex 4 is a fundamental step forward in the Flex architecture that will be a multi-cycle transition
  - Our goal is to introduce the new architecture in a way that is not too confusing for developers
  - Enable them to take advantage of Spark while still allowing them to leverage MX as needed
  - Make both component sets first class citizens in the tool and documentation
  - Spark as default for new users and most use cases
Positioning

- **Who should still use Halo/ MX component set?**
  - Existing users mid-stream in a project, have an investment in MX
  - People currently using Flex who want to stick with what they know
  - Enterprise users who have a longer upgrade cycle

- **Who should use Spark immediately?**
  - Everyone else
    - High-design, high customization
    - Using Catalyst
    - New users in general
Agenda

- Challenges
- Resolutions
- Positioning
  - Additional Features
- Roadmap
- Community Features
Additional Features

- Effects: Better Auto-Centering
- Effects: Bounce and Elastic Easing
- Runtime DesignLayer
- FXG 2.0
- Font2SWF Utility
- Spark Accessibility
- Layout Mirroring
- Strobe Integration
Agenda

- Challenges
- Resolutions
- Positioning
- Additional Features
- **Roadmap**
- Community Features
Roadmap

Late 2009

Flex 4 Beta 2

- MX navigators with Spark controls
- Spark ButtonBar support for MX ViewStack
- ControlBar
- List drag/drop
- More styles for Spark skins
- BorderContainer
- Spark Form

Early 2010

Flex 4.0 Release
Agenda

- Challenges
- Resolutions
- Positioning
- Additional Features
- Roadmap

- Community Features
Open Features

- Spark divider
- IconButton
- Slider Tick Marks
- Runtime FXG parser
- Spark ToolTip
- Spark Alert
- Spark ProgressBar
- RangeSlider (2 thumbs)