Creating Advanced Custom UI Components with Adobe Flex

presented by Maxim Porges

what makes a component “advanced”?

advanced vs. simple

Simple Customizations

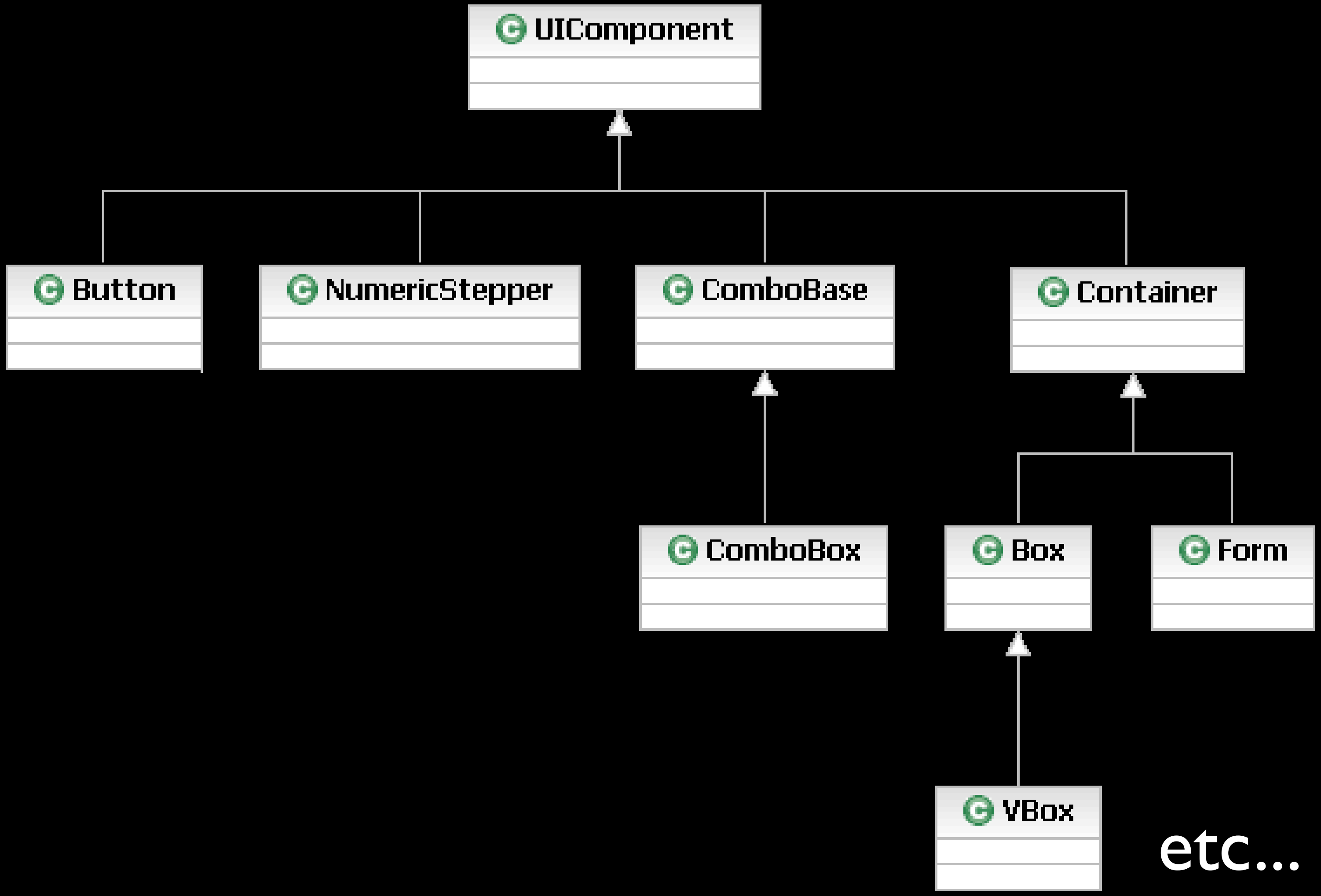
- Add a built-in event handler
- Add custom properties
- Set common behavior for application controls

Advanced Customizations

- Significantly modify visual appearance
- Composite components
- Totally new, direct subclass of UIComponent

UIComponent

class hierarchy



etc...

lifecycle methods

UIComponent protected methods

- createChildren()
- commitProperties()
- measure()
- layoutChrome() **Container subclasses only*
- updateDisplayList()

invalidate* methods

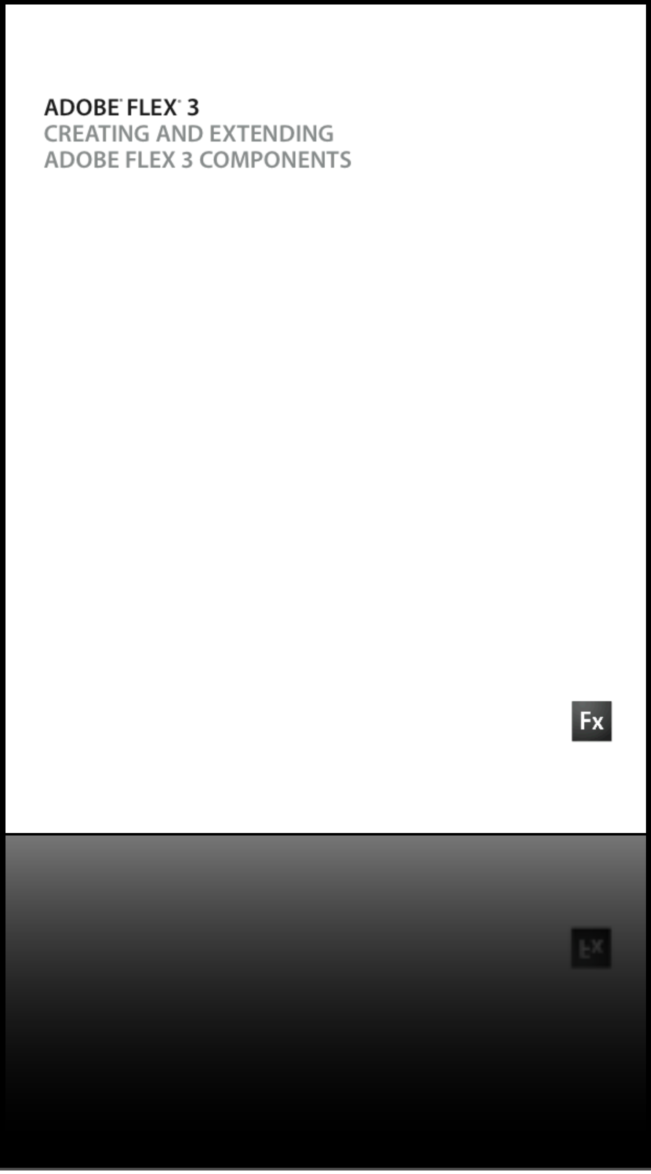
UIComponent invalidation methods

- invalidateProperties()
  Calls commitProperties() on next screen update
- invalidateSize()
  Calls measure() on next screen update
- invalidateDisplayList()
  Calls updateDisplayList() on next screen update

let’s see what they do

learn oodles more

the customization bible



ADOBE® FLEX® 3
CREATING AND EXTENDING
ADOBE FLEX 3 COMPONENTS

ADOBE FLEX 3
CREATING AND EXTENDING
ADOBE FLEX 3 COMPONENTS

beyond the basics

- **Using metadata effectively**
  [Inspectable] *for compiler hints*

  [DefaultProperty] *for adding children in MXML*

- **Declaring/handling custom styles**
  Overriding the styleChanged() *method, using* [Style] *metadata, etc.*

some UI frameworks

- **OpenFlux**
  Very MVC-ish
  http://code.google.com/p/openflux/

- **Degrafa**
  Declarative Graphics Framework
  http://code.google.com/p/degrafa/

thanks :)