

# Rich Symbols in Fireworks CS3

Hands On

Trevor McCauley

Quality Assurance Engineer

Flash Player Quality Engineering

Adobe Systems Incorporated



# About Trevor McCauley



- Currently Flash Player Quality Engineer for Adobe
- Past:
  - Flash/Fireworks Support for Adobe
  - Lead Developer for Pixel Workshop
- Wrote the MXML and Images export for Fireworks CS3
- Web site:  
<http://www.senocular.com>
- Contributor:  
<http://www.kirupa.com>  
<http://www.actionscript.org>  
<http://www.ultrashock.com>  
<http://www.flashkit.com>  
<http://www.fireworksguru.com>

\*A 10 point footnote can go here, if necessary

## Hands On: Fireworks CS3 Rich Symbols Agenda

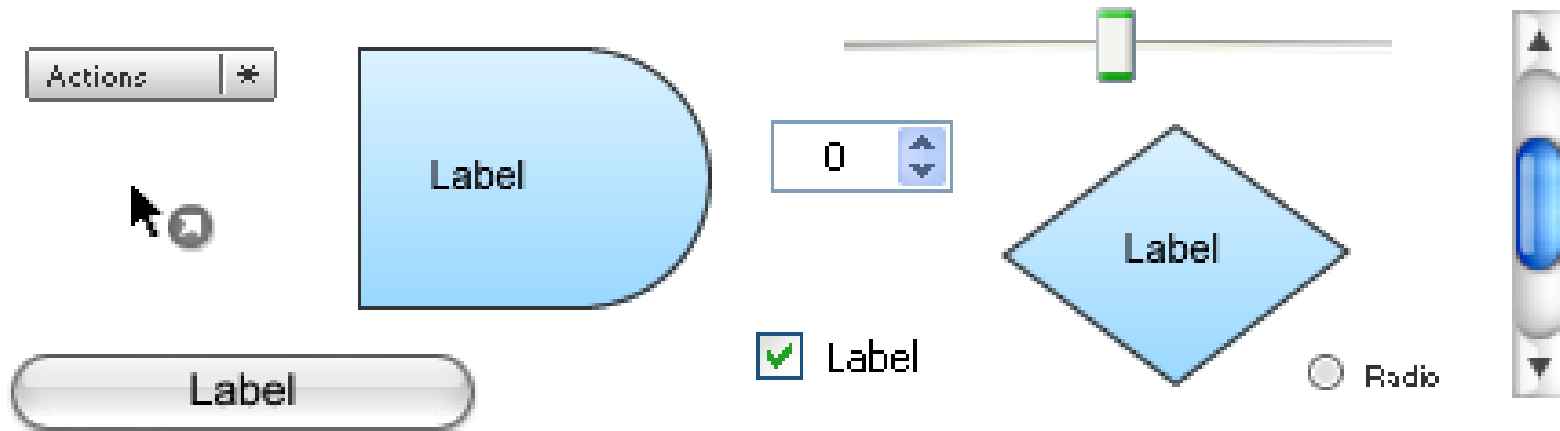
- Using pre-made rich symbols from the Common Library
- Creating new rich symbols using the Create Symbol Script command
- Defining advanced rich symbols by editing JavaScript
- Creating rich symbols that will work with the Fireworks MXML export

## Using pre-made rich symbols from the Common Library

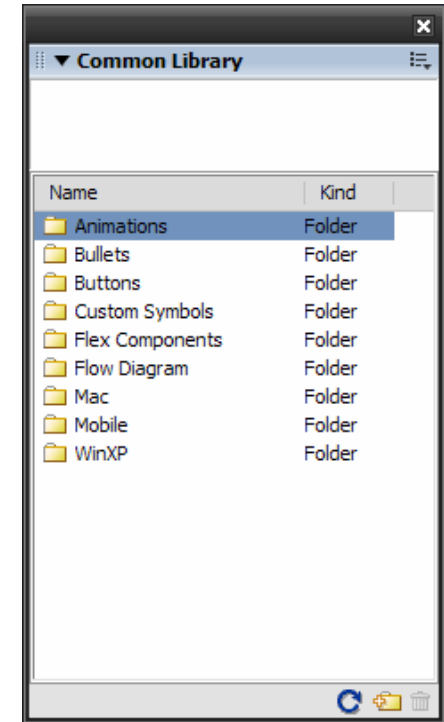
\* Time permitting

# About Rich Symbols

- New kind of graphic symbol in Fireworks CS3
- Advantages
  - 9-slice scaling
  - Instance-specific symbol properties
- Available in the Common Library

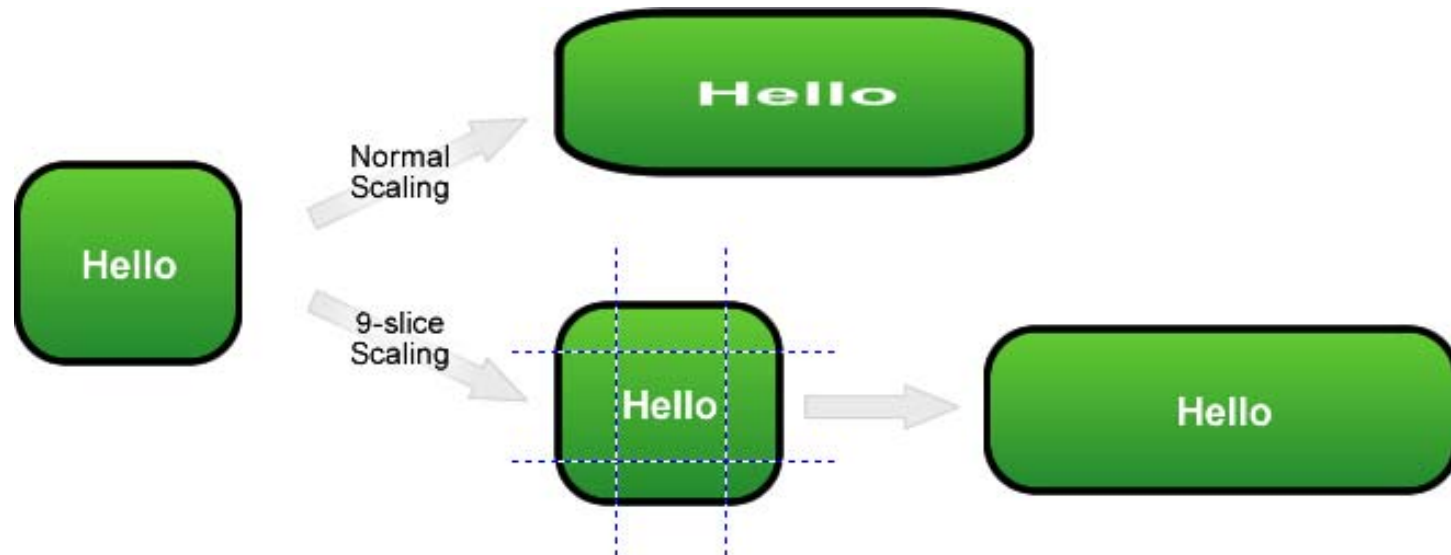


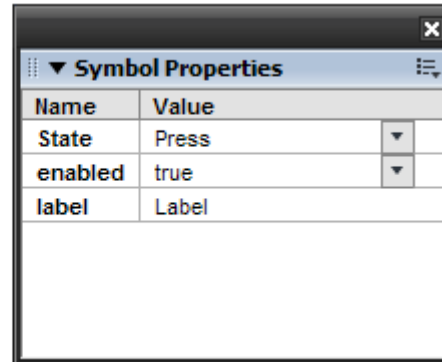
- Symbol repository
- Contains
  - Symbols shipped with Fireworks
  - User's custom symbols
- Symbols from Common Library are added to Document Library



# About 9-Slice "Intelligent" Scaling

- Symbol in a grid of 9 slices
- Slices scale intelligently





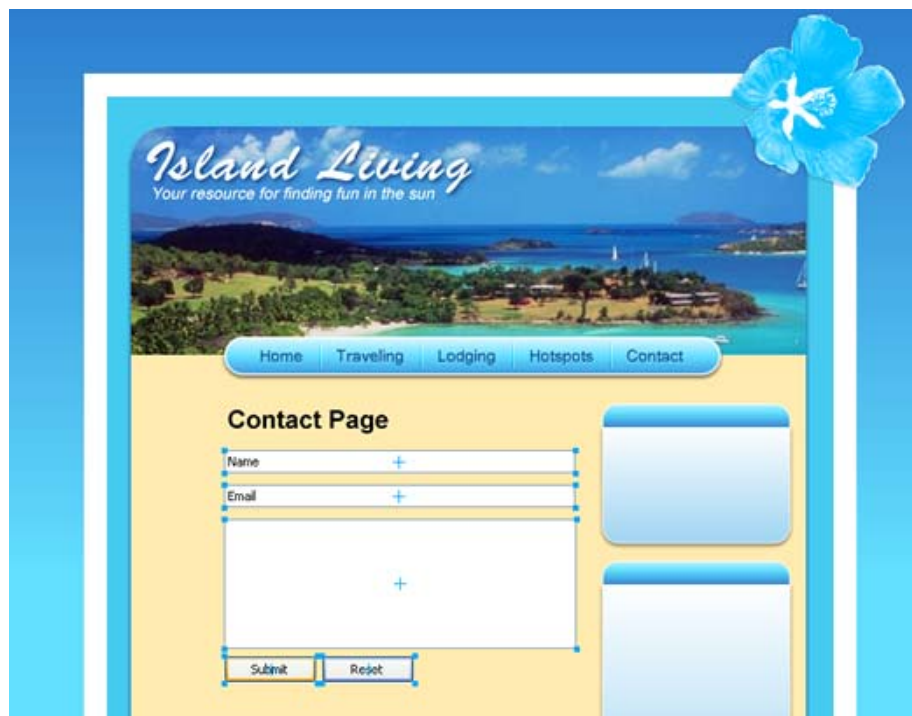
| Name    | Value |
|---------|-------|
| State   | Press |
| enabled | true  |
| label   | Label |

- Change Symbol Characteristics
  - Text, colors, states, and more
- Instance-specific

# Exercise 1: Using Pre-built Rich Symbols



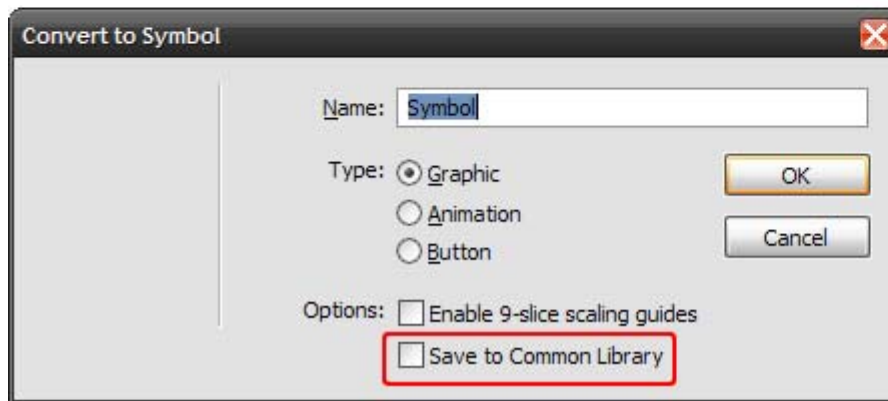
- **Task:** Create a web form prototype for a web site using a pre-existing template
  - Add pre-build rich symbols to a document
  - Modify rich symbol properties



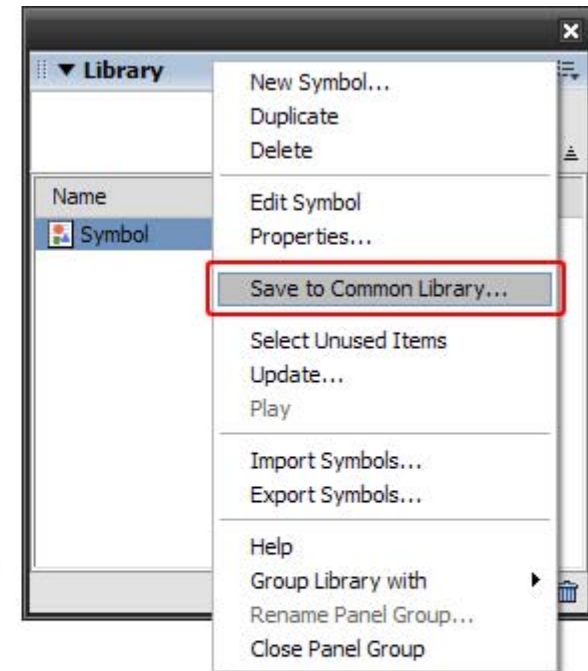
## Creating new rich symbols using the Create Symbol Script command

\* Time permitting

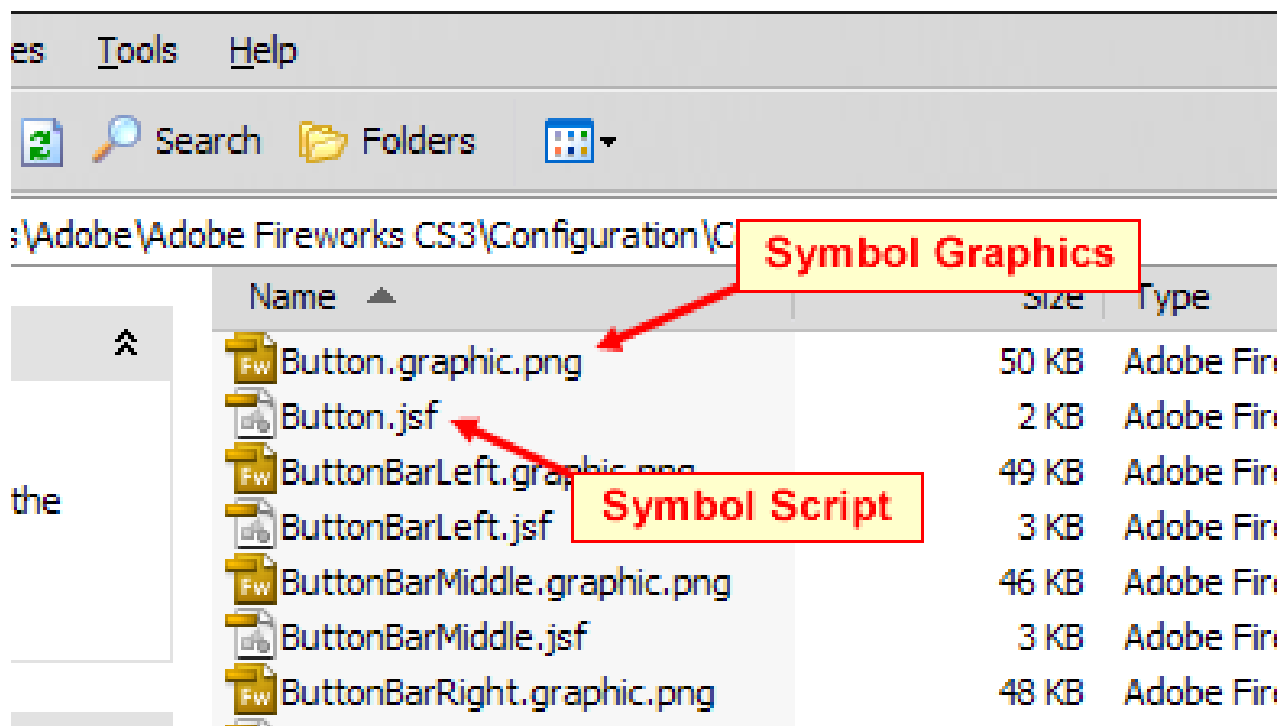
- Saving to the Common Library
  - From New Symbol dialog
  - From existing symbol in the Library



or

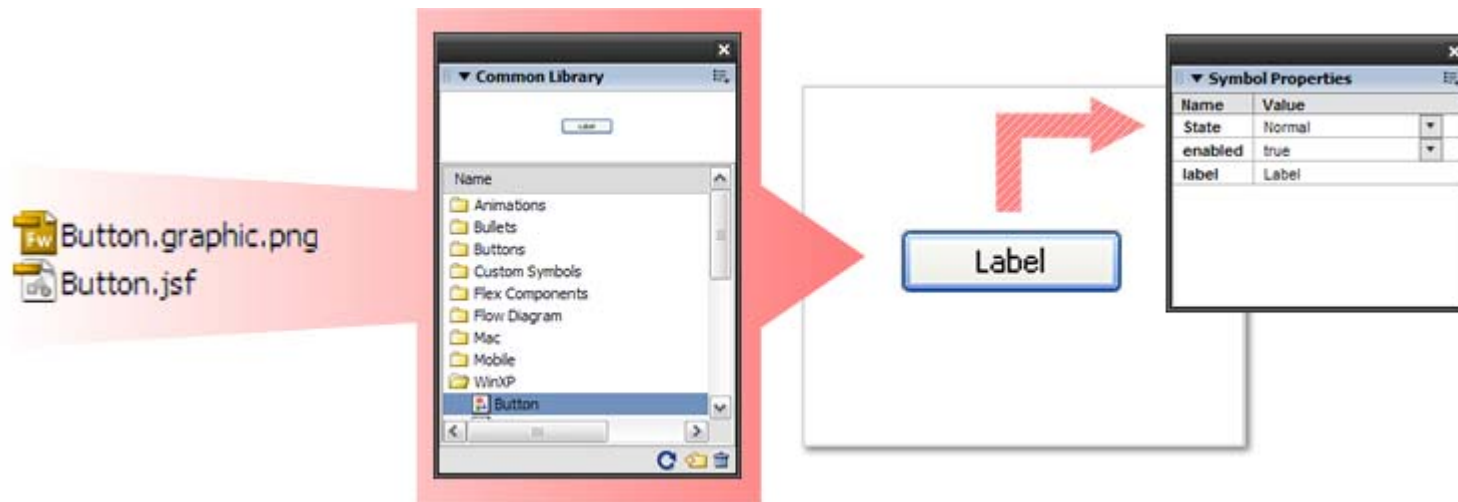


- Symbols are saved as PNG files
- Symbol Properties are defined in JavaScript (.jsf)

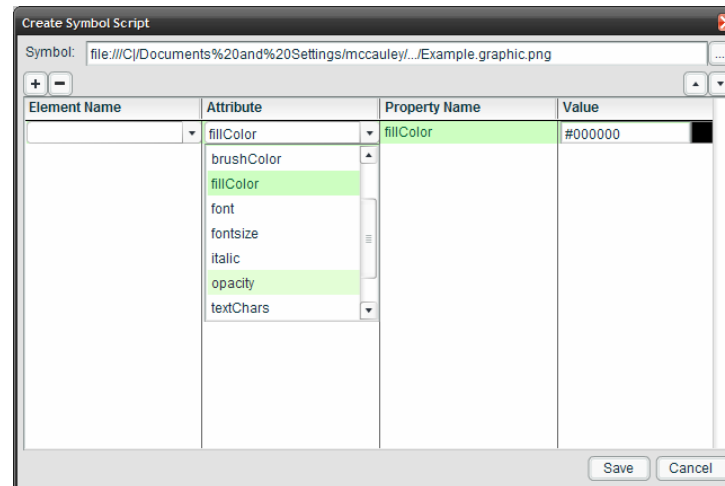


- Symbols in the Common Library represent files on your hard drive
  - Windows:
    - **App:** C:\Program Files\Adobe\Adobe Fireworks CS3\Configuration\Common Library\
    - **User:** <user settings>\Application Data\Adobe\Fireworks CS3\Common Library\
  - Mac:
    - **App:** Macintosh HD/Applications/Adobe Fireworks CS3/Configuration/Common Library/
    - **User:** <user>/Library/Application Support/Adobe/Fireworks CS3/Common Library/

- Symbol script association made when symbol is created from Common Library

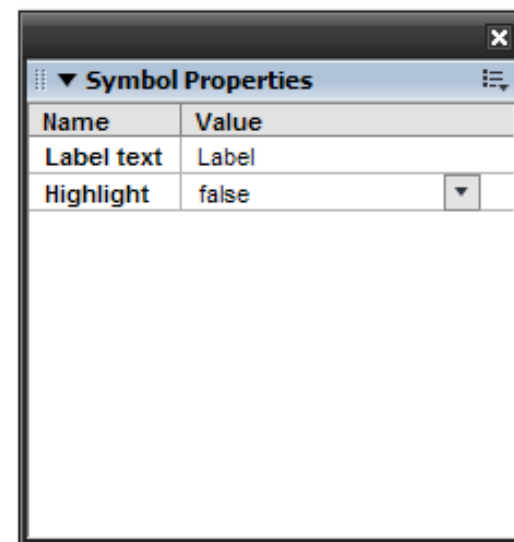
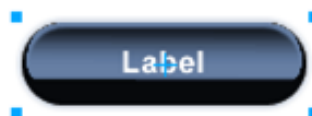
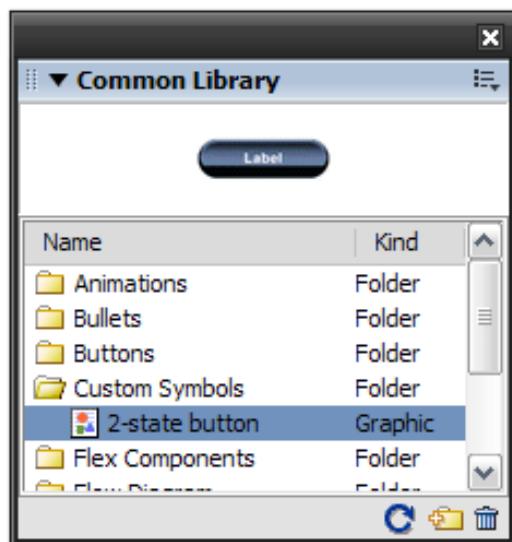


# About Create Symbol Script



- Create Symbol Script Command  
Commands > Create Symbol Script
- Easily define symbol properties
- No JavaScript knowledge needed

- **Task:** Create a rich symbol with symbol properties using the Create Symbol Script Command
  - Create a rich symbol
  - Save it to the Common Library
  - Give the symbol properties using the Create Symbol Script command



## Defining advanced rich symbols by editing JavaScript

\* Time permitting

- JavaScript definition divided in two parts
  - Initialization
  - Updates
- **Initialization:** when a rich symbol is added to the canvas
  - Sets up properties and default values
- **Updates:** whenever values are modified using the Symbol Properties panel
  - Updates/modifies symbol elements

- `Widget.opCode`: indicates initialization or update
  - Initialization: `Widget.opCode = 1`
  - Update: `Widget.opCode = 2`
- `currentValues`: Location to store property definitions
  - `currentValues` definitions each contain values:
    - **name**: Property name in the Symbol Properties panel
    - **type**: Property data type (Properties Panel control)
    - **value**: Default property value
  - Example: `{ name:"Fill Color", type:"color", value:"#ffcc00" }`
- `Widget.GetObjectByName( "name" )`: access to elements in symbols to modify their characteristics
  - Example: `Widget.GetObjectByName("arrow").pathAttributes.fillColor = value;`

- This is the symbol script used for Bullets > Bullet\_Arrow01

```
function setDefaultValues(){
    var values = new Array();
    values.push({ name:"Fill Color", type:"color", value:"#ffcc00" });
    Widget.elem.customData["currentValues"] = values;
}

function applyCurrentValues(){
    var values = Widget.elem.customData["currentValues"];
    Widget.GetObjectByName("arrow").pathAttributes.fillColor =
        values[0].value;
}

switch (Widget.opCode){
    case 1: setDefaultValues(); break;
    case 2: applyCurrentValues(); break;
}
```

## Types

- "color"
- "combobox"
- "fixedlist"
- "list"
- "number"
- "boolean"
- "font"
- "text"

## Format

- "Hex color"
- "selected value,value 1,value 2[,value N]"
- "value 1,value 2[,value N]"
- "value 1,value 2[,value N]"
- "value" or "value,min,max"
- "true" or "false"
- "Font name"
- "Any characters"

## Example

- "#00FF00"
- "A,A,B,C,D"
- "A,B,C,D"
- "A,B,C,D"
- "0,-100,100"
- "true"
- "Arial"
- "Edit this text"

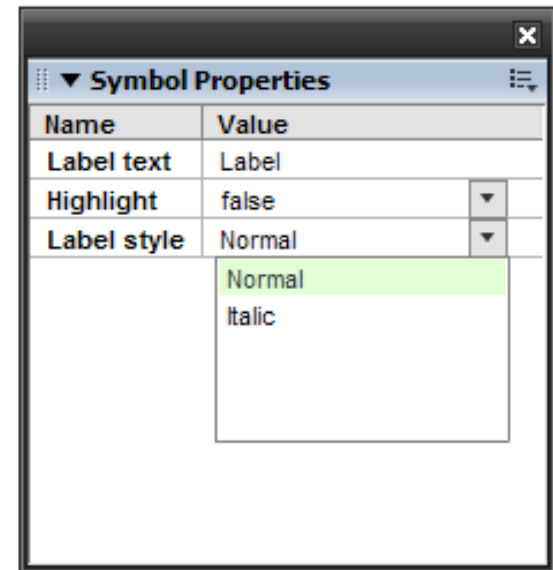
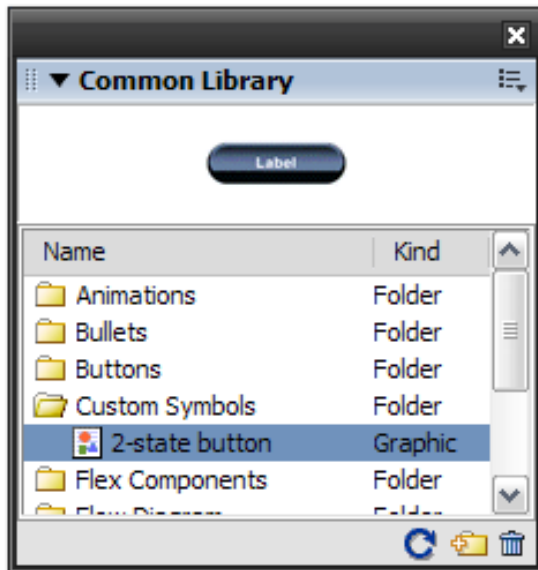
- Rich Symbol element properties usable from `Widget.GetObjectByName()`:

[http://help.adobe.com/en\\_US/Fireworks/9.0/Extending/frw\\_rich\\_symbols\\_rs\\_3.html](http://help.adobe.com/en_US/Fireworks/9.0/Extending/frw_rich_symbols_rs_3.html)

| Attributes                     | Type                                     |
|--------------------------------|--|
| <code>element.left</code>      | integer                                  |
| <code>element.width</code>     | integer                                  |
| <code>element.height</code>    | integer                                  |
| <code>element.pixelRect</code> | <code>rect(left,top,roght,bottom)</code> |
| <code>element.visible</code>   | boolean                                  |

*The list goes on...*

- **Task:** Edit the symbol script for an existing rich symbol
  - Create a new property for an existing rich symbol
  - Edit a rich symbol script using a text editor
  - Create a new instance of the symbol with the new property accessible



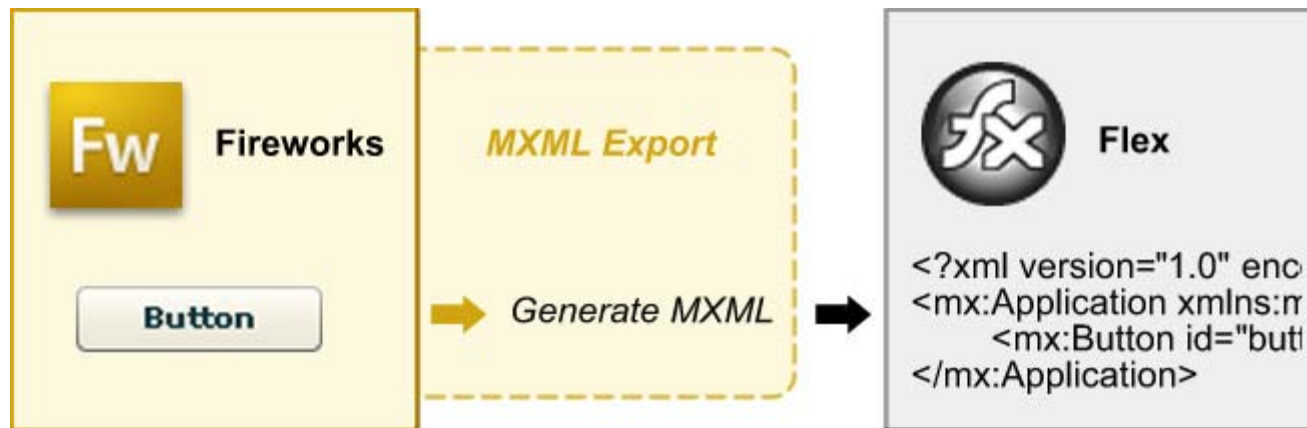
## **Creating rich symbols that will work with the Fireworks MXML export**

\* Time permitting

# About the MXML Export



- Generates content for Adobe Flex Builder 2
- Uses Flex components rich symbols
- Symbol scripts control generated MXML



- MXML Export looks for the following rich symbol properties:
  - `flexClassName`
  - `flexClassDefinition`
- **flexClassName:** The tag name (class name) of the Flex component
- **flexClassDefinition:** Additional Flex-specific properties
  - *Note: flexClassDefinition is handled internally for Flex components shipped with Fireworks*

- This is the symbol script used for Flex Components > ToolTip

```
function setDefaultValues(){
    Widget.elem.customData["flexClassName"] = "ToolTip";
    var currValues = new Array();
    currValues.push({ name:"text", type:"Text", value:"ToolTip" });
    Widget.elem.customData["currentValues"] = currValues;
}

function applyCurrentValues(){
    Widget.GetObjectByName("text").textChars =
        Widget.elem.customData["currentValues"][0].value;
}

switch (Widget.opCode){
    case 1: setDefaultValues(); break;
    case 2: applyCurrentValues(); break;
    default: break;
}
```

## Types

- margin
- padding
- sizeOffset
- customValues
- attributeProperties
- styleProperties
- defaultProperties
- namespace
- textOnly
  
- ignored

## Format

- Rectangle
- Rectangle
- Object (width, height)
- Object
- Array
- Array
- Array
- Object
- Text
  
- Boolean

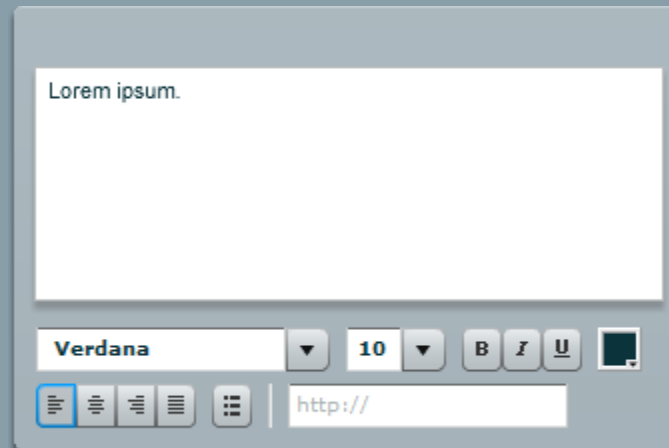
## Example

- {top:5, right:5, bottom:5, left:5}
- {top:5, right:5, bottom:5, left:5}
- {width:10, height:10}
- {myNum:10, click:"clickHandler();"}
- ["label", "enabled"]
- ["color", "textSelectedColor"]
- ["x", "y", "width", "height"]
- {ns:"http://www.example.com"}
- "<mx:ColorPicker id=\"cp\" selectedColor=\"0xFFFFFFFF\"/>"
  
- true

# Exercise (Demo) 4: MXML-ready Symbol



- **Task:** Create a rich symbol for use with the MXML export
  - Create a Flex component symbol in Fireworks
  - Create a symbol script allowing it to export to MXML



- When creating new symbols, do **not** copy or reuse existing rich symbol symbols or graphic PNGs. An internal identifier associates symbol instances with their source. Copies retain that same identifier so the copied source will be indistinguishable from the original
- To save yourself from editing a symbol for object naming, you can name objects used in a symbol on your main canvas saving the symbol directly to the common library when you convert those objects into a symbol using `F8` or `Modify > Symbol > Convert to Symbol`
- For creating new symbol scripts, the Create Symbol Script command can be a great way to quickly create a template, even if you plan to use more advanced code in the script
- Try your best to foresee all necessary properties required for rich symbols you create to prevent the need to update them later

- Creating a rich symbol with 9-slice and 3-slice scaling *by Tom Green*  
[http://www.adobe.com/designcenter/video\\_workshop/html/vid0175.html](http://www.adobe.com/designcenter/video_workshop/html/vid0175.html)
- Creating property scripts for rich symbols *by Tom Green*  
[http://www.adobe.com/designcenter/video\\_workshop/html/vid0173.html](http://www.adobe.com/designcenter/video_workshop/html/vid0173.html)
- Using rich symbols in Fireworks *by Aaron Beall*  
[http://www.adobe.com/designcenter/fireworks/articles/frwcs3\\_richsymbol.html](http://www.adobe.com/designcenter/fireworks/articles/frwcs3_richsymbol.html)
- Fireworks Documentation:  
<http://www.adobe.com/support/documentation/en/fireworks/>

- Exporting MXML and Images from Fireworks CS3 by *Trevor McCauley*  
[http://www.adobe.com/devnet/fireworks/articles/mxml\\_images\\_export.html](http://www.adobe.com/devnet/fireworks/articles/mxml_images_export.html)
- Creating rich symbols for MXML and Images export in Fireworks CS3 by *Trevor McCauley*  
[http://www.adobe.com/devnet/fireworks/articles/rich\\_symbols\\_export.html](http://www.adobe.com/devnet/fireworks/articles/rich_symbols_export.html)
- Creating a custom rich text editor Flex 2 component with Fireworks CS3 by *Darrell Heath*  
[http://www.adobe.com/devnet/fireworks/articles/custom\\_rich\\_text\\_editor.html](http://www.adobe.com/devnet/fireworks/articles/custom_rich_text_editor.html)
- Fireworks documentation on creating rich symbols for MXML:  
[http://help.adobe.com/en\\_US/Fireworks/9.0/Extending/frw\\_rich\\_symbols\\_rs\\_5.html](http://help.adobe.com/en_US/Fireworks/9.0/Extending/frw_rich_symbols_rs_5.html)

A large, light gray question mark is centered on the page, serving as a background for the title.

## Questions and Answers

# Rich Symbols in Fireworks CS3

