YAHOO! FLASH PLATFORM

Item Renderers in Flex

Josh Tynjala
Flash and Flex Engineer, Yahoo!
360Flex Atlanta

YAHOO! FLASH PLATFORM

Yahoo! Flash Platform

Developer Network Home Help

Search the site

YAHOO! DEVELOPER NETWORK

YDN → Flash Developer Center

FLASH DEVELOPER CENTER

Welcome Flash Developers

The **Yahoo! Flash Developer Center** is an ever growing collection of Flash tools, resources and information — all open source and freely available for your use. Check back often for updates, and join the ydn-flash Yahoo! Group to exchange information with Flash developers just like yourself.

Yahoo! Flash Blog

ASTRA Galore: New Flash and Flex Components
01/30/08, at 02:01 PM, by Allen Rabinovich

Hello, My Name Is Badge Kit
12/20/07, at 11:12 PM, by Alario Cole

Yahoo! and Flex: Feel Good In Your Own Skin
12/13/07, at 03:12 PM, by Allen Rabinovich

YUI Got a Little More Flashy Today!
12/04/07, at 06:12 PM, by Josh Tynjala

ASTRA Aquarium: A Fun and Instructive Example Application
11/26/07, at 11:11 AM, by Michael Hoch

YDN-Flash Yahoo! Group

Join the ydn-flash group to connect to other developers like you and send comments and suggestions (and even bug reports!) to Yahoo!'s Flash Platform team.

Adding new Context Menu Entry in Right-Click Context Menu in AS3 Yah
02/04/08, at 12:02 AM, by Arun M

Re: Problem to plot Line Chart
02/03/08, at 11:02 PM, by Josh Tynjala

ASTRA: ActionScript Toolkit for Rich Applications

ASTRA, the ActionScript Toolkit for Rich Applications, is a collection of Flash and Flex components, code libraries, toolkits and utilities developed by Yahoo! for ActionScript developers. These libraries available under the terms of the open source BSD license.

Flash Components

Yahoo! offers the ASTRA Flash components to complement the existing set provided in Adobe Flash CS3. These components are easily skinnable and fully documented, and wide selection of examples are available to help developers get started quickly.

Components:
- Tree
- Menu
- TabBar
- AutoComplete
- Charts
- AlertManager
- AudioPlayback
- MenuBar

Resources:
- Overview
- API Documentation
- License

DOWNLOAD

Flex Components

Yahoo! offers the ASTRA Flex components for use in your Adobe Flex 3 applications. These Flex components are designed to work like any other components for Flex with familiar properties, styles, and events. They're fully documented, and each component comes with multiple examples that demonstrate its various uses.

Components: Resources:

Featured Application: Aquarium

A fun application using ASTRA components.
More: Aquarium

Flash Theater

Play Video

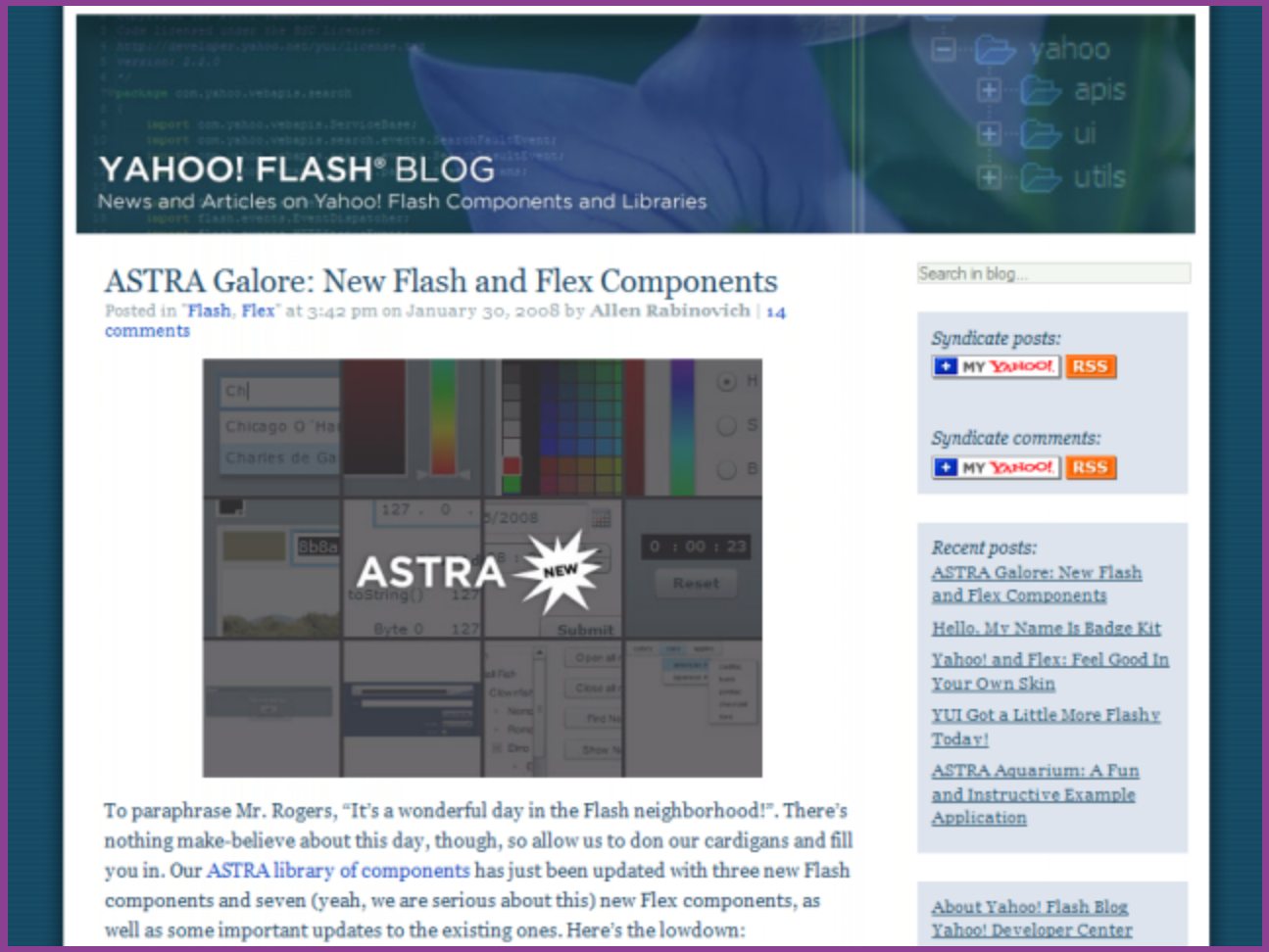
VIDEO

A screencast on ASTRA components.
More: Charts, AutoComplete and TabBar

Articles

http://developer.yahoo.com/flash/

Yahoo! Flash Blog



http://www.yswfblog.com/

Zeus Labs

zeuslabs

JOSH TYNJALA EXPLORES FLASH, FLEX, AND ACTIONSCRIPT

Juice Analytics Features Flex TreeMap Component

JANUARY 31, 2008

Today, I noticed my TreeMap component for Flex got featured by Juice Analytics on their blog. They created an application that combines a treemap with comScore data and Alexa thumbnails to make an interesting little visualization of Internet traffic.

At a high level, the application shows what categories of websites are the most popular online (based on a comparison of unique visitors between August and November 2007). Digging deeper, one can discover other mysteries across the categories. For instance, you might notice that various Yahoo! properties are sprinkled throughout the treemap because, obviously, the company is a major Internet portal that covers a wide-variety of interests. Disclaimer: I work for Yahoo! and I'm not ashamed to point out cool stuff about my employer. By the way, did you see the Flex components we released yesterday? 😉

About the Author

WHO THE HECK IS THIS GUY?

Josh Tynjala is a professional ActionScript Engineer living in Silicon Valley. He works for Yahoo! where he develops Flash and Flex components.

Recommendations

GOOD FLASH BOOKS THAT I'VE READ

Essential ActionScript 3.0
by Colin Moock

ActionScript 3.0 Design Patterns
by Sanders and Cumaranatunge

Search

CAN I HELP YOU FIND SOMETHING?

GO

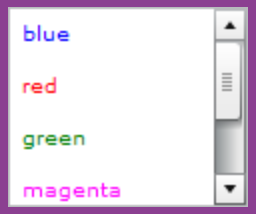
Recent Posts

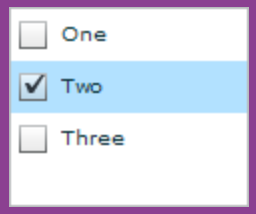
http://www.zeuslabs.us/

Agenda

- Custom Renderers for Existing Components
- Build an Item Renderer System

Custom Renderers for Existing Components





Setting Custom Renderers in MXML

- Option 1

```
<mx:List itemRenderer="mx.controls.Label">
```

- Option 2 (more in a bit)

```
<mx:List>
    <mx:itemRenderer>
            <mx:Component>
                    <mx:Label/>
            </mx:Component>
    </mx:itemRenderer>
</mx:List>
```

Setting Custom Renderers in ActionScript

- mx.core.IFactory
- mx.core.ClassFactory

```
import mx.controls.Label;

itemRenderer = new ClassFactory( Label );
itemRenderer.properties = { minWidth: 400 }

var instance:IListItemRenderer =
                itemRenderer.newInstance();
```

Creating Custom Renderers in MXML

- Uses special <mx:Component> tag.
- Advantages:
  - No extra classes.
  - Easy MXML layout.
- Disadvantages:
  - Can be hard to maintain.
  - Must understand scope.

MXML Item Renderer Scope

```
<mx:Array id="dp">
   <mx:String>One</mx:String>
   <mx:String>Two</mx:String>
   <mx:String>Three</mx:String>
</mx:Array>


<mx:List dataProvider="{dp}">
   <mx:itemRenderer>


      <mx:Component>
            <!-- this is like a new MXML file! -->
            <mx:CheckBox/>
      </mx:Component>

   </mx:itemRenderer>
</mx:List>
```

Using outerDocument in <mx:Component/>

- Refers to scope of MXML outside the renderer.

```
<mx:Component>
   <mx:CheckBox
       change="trace(outerDocument.dp.indexOf(this.data))">
   </mx:ComboBox>
</mx:Component>
```

Creating Custom Renderers in ActionScript

- Implement the required interface
  - IListItemRenderer
  - Used by List, Tree, and DataGrid
- Optionally, implement the drop-in interface
  - IDropInListItemRenderer

Drop-In Renderers

- A common data structure
- Promotes encapsulation
- The standard Flex components are ready for drop-in

ListData Class

- The common data structure
- Properties
  - columnIndex
  - rowIndex
  - icon
  - label
  - labelField
  - uid
  - owner

Custom Renderer Examples

- Font List (MXML)
- Color List (MXML)
- CheckBox List (ActionScript)

Build an Item Renderer System

Expose Common Renderer Properties

- Examples from List:
  - label
  - icon
- Fields and Functions
  - Field is property in data provider item.
  - Function is custom procedure to "generate" the value.

Create Quick Conversion Functions

- Examples from List
  - itemToLabel()
  - itemToIcon()
  - itemToDataTip()
  - itemToItemRenderer()
  - itemRendererToIndex()

itemToLabel() Example

```
public function itemToLabel( item:Object ):String
{
   if(!item) return "";

   if( this.labelFunction != null )
   {
       return this.labelFunction( item );
   }
   else if( item.hasOwnProperty( this.labelField ) )
   {
       return item[ this.labelField ];
   }

   // worse case scenario
   return item.toString();
}
```

Creating, Removing, and Updating

- Reduce display list manipulations
- Reuse renderers with new data
- Recycling is fun

Structure of Renderer Creation

```
override protected function commitProperties():void
{
   super.commitProperties();

   // save the current item renderers for reuse.
   this.createCache();

   // reuse or create new renderers to display the data.
   this.renderItems();

   // remove extra cached item renderers we don't need.
   this.clearCache();
}
```

Renderer Recycling

- createCache()
- renderItems()
- clearCache()

Creating the Cache

```
protected function createCache():void
{
   // copy the Array using concat() and start fresh
   this._rendererCache = this._itemRenderers.concat();
   this._itemRenderers = [];
}
```

Renderer Recycling

- createCache()
- renderItems()
- clearCache()

Requesting a Renderer

```
protected function renderItems():void
{
   var iterator:IViewCursor =
       this._dataProvider.createCursor();


   while( !iterator.afterLast )
   {
       // request a renderer. it may be new or from
       // the cache. we don't care.
       var renderer:ItemRenderer = this.getRenderer();
       renderer.data = iterator.current;
       iterator.moveNext();
   }
}
```

Using the Cache

```
protected function getRenderer():void
{
   // if there's anything in the cache, use it
   if( this._rendererCache.length > 0 )
   {
       // shift() removes the first item
       // may help to prevent extra redraws
       return this._rendererCache.shift();
   }
   else // otherwise create a new one
   {
       return this._itemRenderer.newInstance();
       // may want to add event handlers too
   }
}
```

Renderer Recycling

- createCache()
- renderItems()
- clearCache()

Clearing the Cache

```
protected function clearCache():void
{
   var cacheLength:int = this._rendererCache.length;
   for( var i:int = 0; i < cacheLength; i++ )
   {
       // remove the renderer from the cache
       var renderer:ItemRenderer =
       this_rendererCache.pop();

       // then (finally) remove it from the display list
       this.removeChild( renderer );

       // also, remove event handlers, if needed
   }
}
```

Examples

- Flex TreeMap
- Flash CS3 TabBar (I know, I know…)

Yahoo! Flash Platform

- Yahoo! Flash Developer Network:
  - http://developer.yahoo.com/flash/
- Mailing List:
  - http://tech.groups.yahoo.com/groups/ydn-flash/
- Blog:
  - http://www.yswfblog.com/