

INTO THE new  
**BREW 2007 CONFERENCE**

## Adding Animations to UIs

Ken Davis, Staff Technical Training  
Specialist  
QUALCOMM Internet Services





# Agenda

- Introduction
  - Some basics of animation sequences and TrigML®
  -
- Basic Animation Elements
  - Discuss tags that are used by TrigPlayer to render animation effects
- Animating Icons
  - Using animations on images to create interest in menus
- Creating Tickers
  - Using the basics learned above to create scrolling tickers and moving text
- Panels and Animated Menus
  - Setting up panels that move
- Q & A and Wrap-up



# Introduction

- TrigML is a hybrid markup and basic scripting language
  - Markup language provides for presentation of data resources
  - Simple scripting capability allows for sequencing tasks such as with animations
- Various methods for scripting visible elements
  - Visible elements can be animated using the <anim> tag
  - Animated GIFs can be played using the animate attribute of the <image> tag
  - Other elements, such as <slide>, can also be used
  - Multiple <anim>, <slide> or other elements can be sequenced or synchronously triggered
- Animations are triggered by events (`_entry`, `_focus`, etc.) or by conditions (`while`, `until`) of the parent



# Common Attributes of Visible Elements

- These attributes can be animated
  - Position attributes for positioning relative to parent
  - Size attributes of width and height
  - Color (bgcolor and bdcolor)
  - Border attributes (bdwidth, bdstyle, etc.)
  - Other attributes, depending on element, may be available for animation
- Visible elements include conditional attributes that can trigger or terminate events
  - while = {condition} triggers the element while condition is true
  - until = {condition} terminates the element



# Basic Animation Elements

- `<anim>`
  - Child element to a visible parent
  - Modifies an attribute of the parent through a given list of values
  - Animation cycle is performed in a specific duration
- `<slide>`
  - Based on `<anim>` tag but simplified
  - Provides a movement path through a parent's frame
  - Animation cycle is performed in constant rate
- Use these common attributes as triggers
  - `when` = the event that triggers the animation
  - `until` = event used to terminate the animation
  - `while` = event or condition that triggers the element



## <anim>

- Constant time moving animation for a visible element parent, such as an image
- Manipulates an attribute by name, such as x coordinate, through a list of values to perform the animation
- Different attributes of the parent can be animated by using a separate <anim> for each one
- Triggered by an event such as `_focus`, `_keypress`, `_entry`, or some user event



## <anim> Attributes

- name = attribute of parent element to be modified
- value = list of values to animate through
  - Must be of same type as the attribute named
- duration = amount of time for one animation cycle
  - If zero then the end value is immediately applied
  - Expressed in milliseconds (ms)
- repeat = number of times animation is repeated
  - If positive number, animation will be played a total of <repeat + 1> time(s)
  - A negative number specifies infinite number of repeats
  - Zero specifies no repeats and the cycle happens only once
- holdlast = controls how the animation ends
  - If `_true` (default) the <anim> element holds the last value until completion event occurs
  - If `_false` the <anim> element switches off in duration ms after trigger event or after repeats



## Emulating Inertia and Acceleration

- Animation of x and y coordinates give impression of movement
  - Use two <anim> elements to animate both x and y coordinates at the same time
- Non-linear values can suggest inertia and acceleration
  - Gravity and bounce  
<anim name="x" value="0;5;15;30;27;30;29;30"/>
  - Acceleration  
<anim name="x" value="0;2;4;8;16;32"/>

## <anim> Example

```
<image res="images/ball" x="0" y="0">  
  <anim when="_entry" duration="500" name="y"  
    value="0;10;20;40;50;90;120" repeat="0"/>  
  <anim when="_entry" duration="1000" name="x"  
    value="0;10;15;20;25;30;40;50" repeat="0"/>  
</image>
```

- The image of a ball will animate on both x and y coordinates over a list of values, giving the impression of it bouncing across the screen

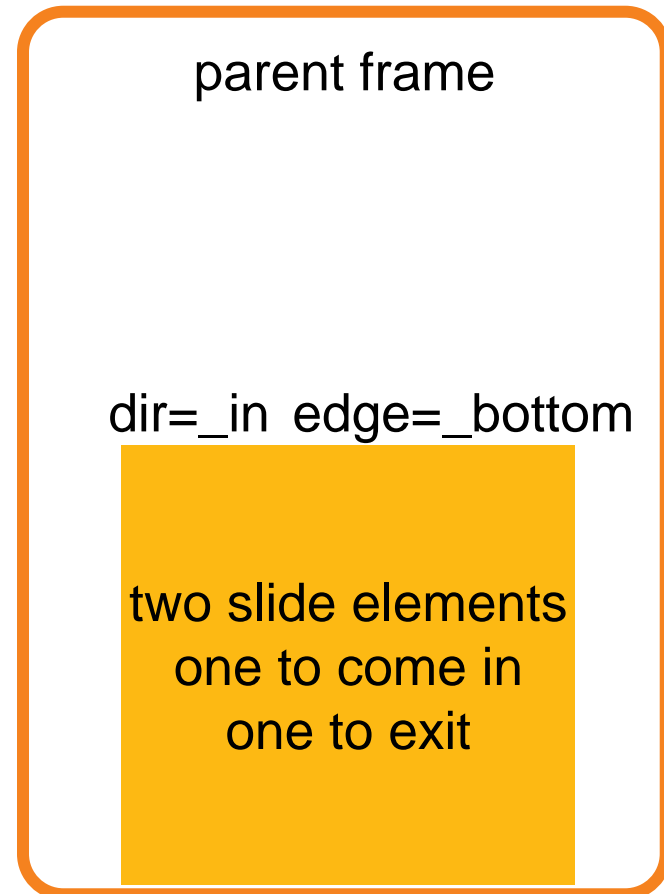


## <slide>

- Based on <anim> but simplified
  - Defines where it comes into the frame, direction of movement, and pace
- Constant rate moving animation for element entering or leaving a parent's frame
  - stepsize attribute provides amount of movement in pixels per step
  - steptime attribute provides the amount of time for each step
- Provides a movement path through the parent by providing an entry position and a direction
  - edge attribute defines entry position into parent frame
  - dir attribute provides direction of movement

## <slide> Attributes

- edge = determines edge for entrance or exit
  - \_left, \_right, \_top, \_bottom
- dir = direction of the animation
  - \_in, \_out
- stepsize = number of pixels to move in each step of the animation
  - Positive integer
  - Not modifiable after set
- steptime = ms between each animation step
  - Not modifiable after set
  - Default is 100



dir=\_out edge=\_bottom

## <slide> Example

```
<seq when="_entry" repeat="-1">  
  <slide edge="_bottom" dir="_in" stepsize="2" steptime="50"/>  
  <throw event="myWait" delay="2000" />  
  <slide edge="_left" dir="_out" stepsize="2" steptime="50"/>  
  <throw target="myTicker" event="_advance"/>  
</seq>
```

- This code sample takes a popup window and slides it up from the bottom of the screen.

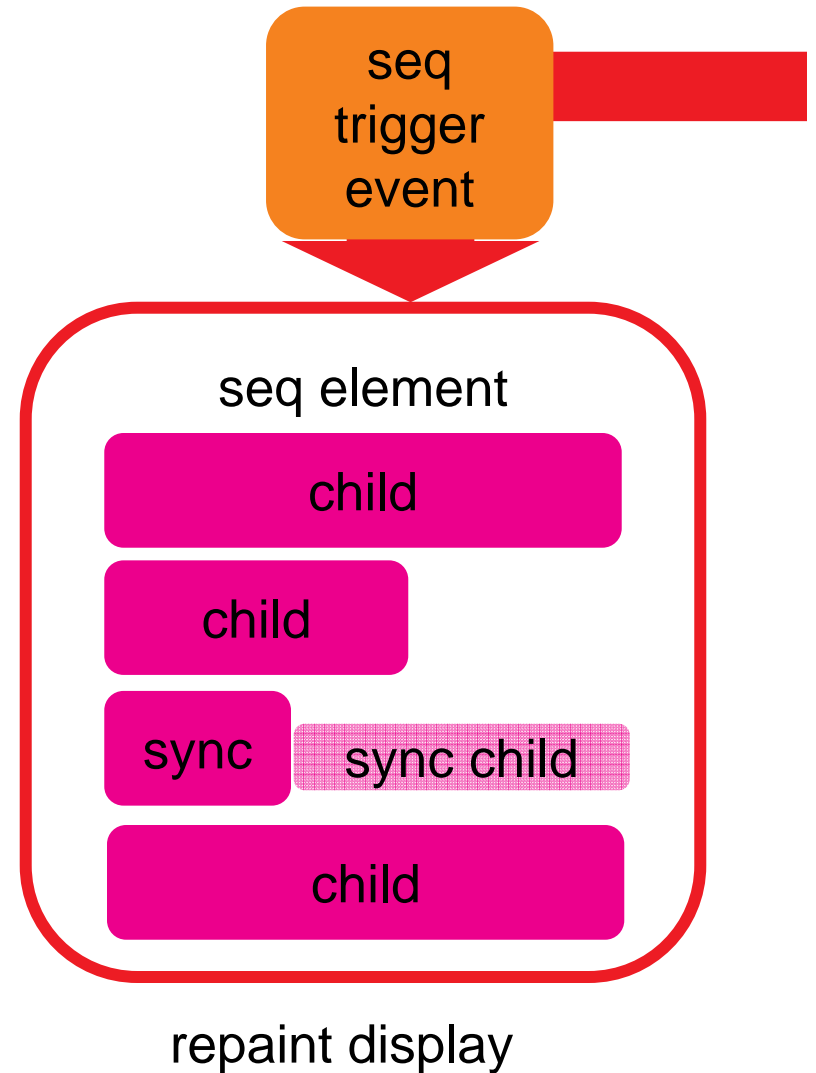


## Other Related Elements

- `<seq>` - to trigger a sequence of elements on an event
  - For a news ticker, the movement into the frame, a pause, then movement out could be a sequence
- `<sync>` - to synchronously trigger a group of elements on an event
  - Have an icon grow in size and move up and down to show focus could be triggered as a synchronous animation

## <seq>

- Sequence of consecutive listening elements
  - Sequence is started when <seq> triggered
  - Each element in the <seq> is triggered when the previous element has finished
  - If a <sync> element is included it is not finished until all of its contained listeners have finished
  - Display repainted, if necessary, after each child element is finished





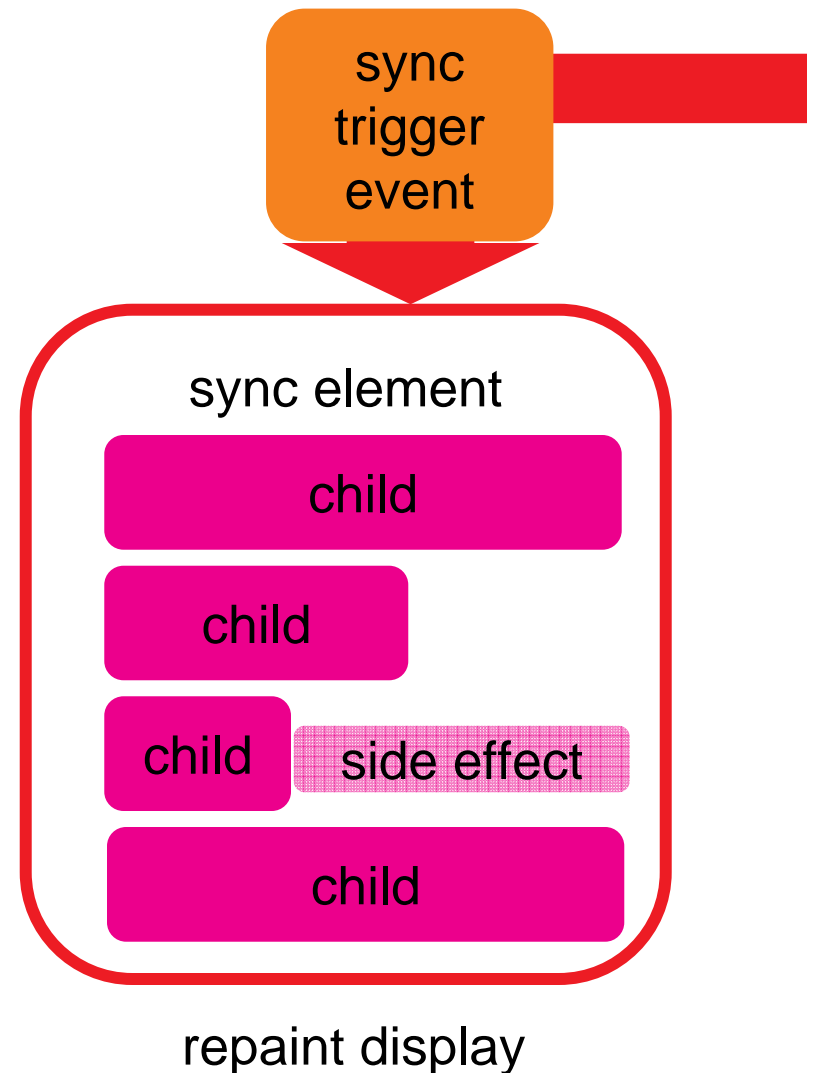
## <seq> Example

```
<seq when="MyEvent" consume="_true">  
  <anim name="y" value="200;-195;-185;-195;200"/>  
  <throw target="item0" event="_advance"/>  
</seq>
```

- This sample code fragment performs an animation on the y coordinate of a parent element and, when complete, throws an \_advance event

## <sync>

- Collection of listening elements triggered simultaneously
  - Visually synchronous action
  - Simultaneous execution when <seq> trigger event occurs
  - Child elements are triggered in the order they are defined
  - Some elements may finish before others
  - Display is not repainted until all children have been triggered and side effects finished





## Animating with animate Attribute

- The animate attribute can be used to automatically animate the frames of a multi-frame image
- animate=<value> determines the number of cycles
  - `_none` = animation does not occur
  - `_once` = animation cycles one time, ending on last frame
  - `_cont` = continuous animation
- `steptime = <time in ms>` determines speed of animation
  - Animated images (e.g., natively animated GIFs) don't require `steptime` as animation is built in
- The animate attribute is switched off by default
  - Must be explicitly switched on
  - Controlled through event trigger or `while/until` conditions



## Throwing Events (A quick review...)

- Listening elements are aware of events, such as “\_focus” indicated by the when attribute

```
<anim when="_focus" name="y"
      value="10;20;30;20;10" />
```
- The developer can create “user” events, such as “myevent” by using <throw>

```
<throw event="myevent" />
```
- An element can listen for this event and react accordingly

```
<anim when="myevent" name="color"
      value="#FFFFFF;#FF000000" />
```
- All elements in the tree will “hear” the event, but only those listening for it will actually do something



## Throwing Events (continued)

- Throwing an event with a delay can be used to create a pause, such as a popup window

```
<throw event="mywait" delay="2000" />
```

- Events can also be thrown to specific listeners by setting a target

```
<throw when="_entry" event="init"  
  target="MainMenu">
```



# Animating Menu Icons



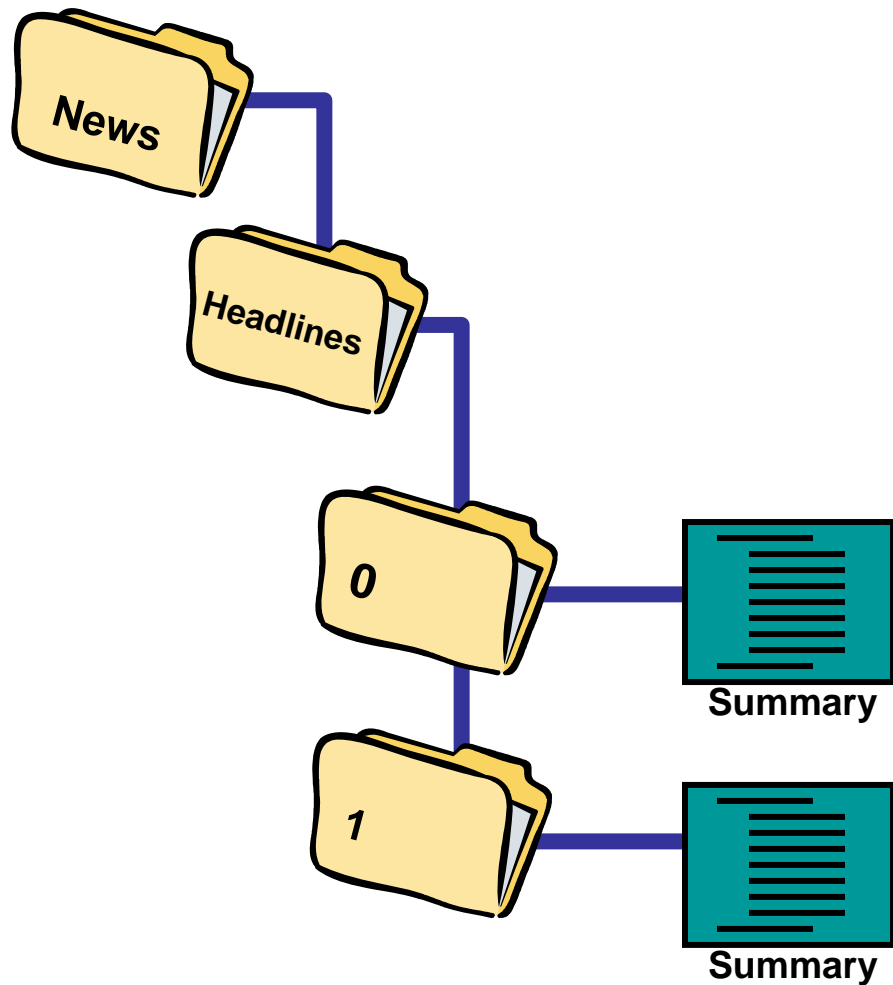
## Menu Animation Demonstration

- Single frame icons
  - Trigger on `_focus` event to animate the icon
  - Ballooning and bouncing effects emulated by use of multiple `<anim>` elements
- Multiple frame icons
  - Use `animate` attribute of image tag to animate icons even without focus
  - Draw attention to high-value services



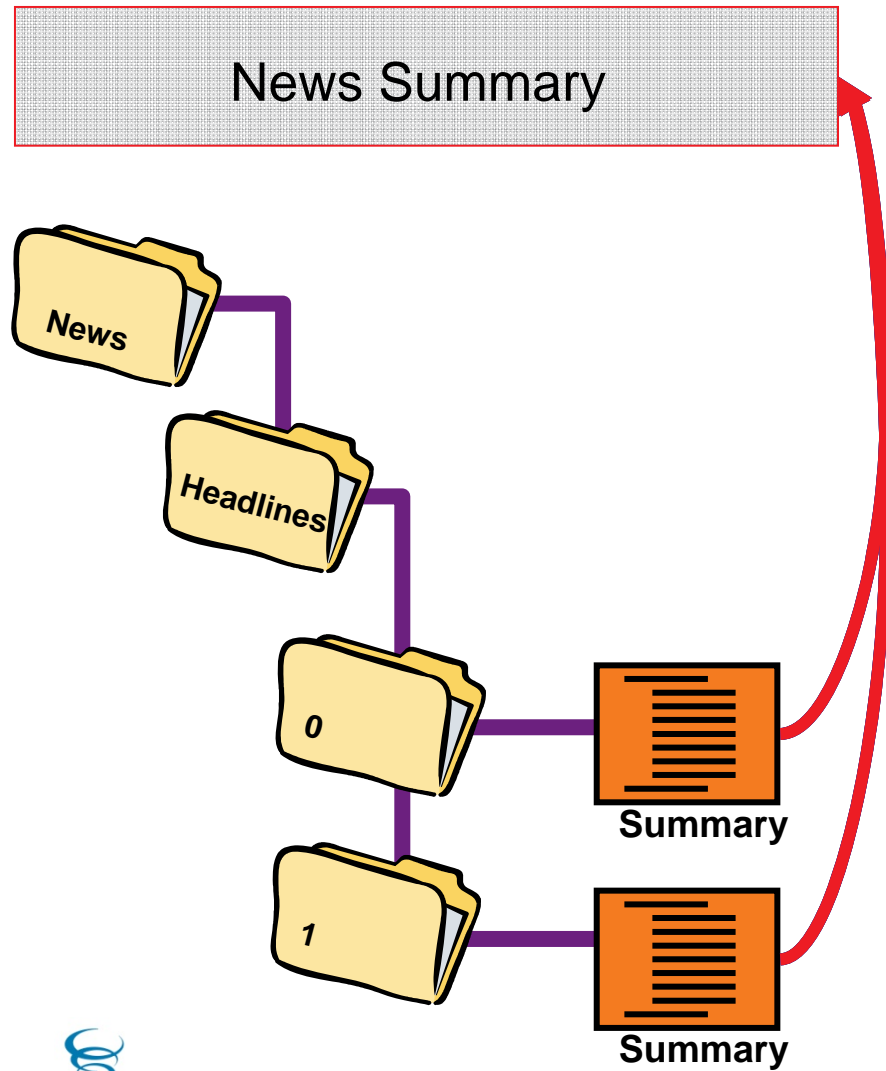
# Creating a News Ticker

# Building a News Ticker



- Assume the following:
  - News stories are provided as a series of text resources
  - Stories are stored in the Headlines folder, with each story in a sequentially numbered folder
  - There is at least one story, and possibly several stories
  - Each story needs to be read in a scrolling ticker

# Building a News Ticker (continued)



- Desired results
  - To iterate through each folder, grab the story and display it in a ticker
  - This process will repeat indefinitely



## Build the Ticker

- Create text resource place holders for the text that will appear in the ticker and note the path
- Create the ticker template
  - Create a group to hold the ticker (MyTicker)
  - Create a group for the ticker itself (AnimGroup)
  - Use repeatover attribute for this group and point it to the resource path from above
  - Create a <seq> element to hold the animation and set repeat=-1 to repeat indefinitely
  - Use <slide> to slide the text into the frame and then slide out
  - Throw a dummy “MyWait” event with a delay to pause
  - Throw an \_advance event to the MyTicker to advance the repeatover iterator

# Constructing the Ticker Template

```
<group id="myTicker" w="176" h="20" y="34" bgcolor="#FF808080"
broadcast="_true" repeatover="news/headlines">
  <group id="AnimGroup" h="20" w="-" x="_left"
  broadcast="_true">
    <seq when="_entry" repeat="-1">
      <slide edge="_bottom" dir="_in" stepsize="2" steptime="50"/>
      <throw event="myWait" delay="2000" />
      <slide edge="_left" dir="_out" stepsize="2" steptime="50"/>
      <throw target="myTicker" event="_advance"/>
    </seq>
    <text y="0" color="#FFFFFFFF" res="news/headlines/$$/summary" >
      <att when="_focus" name="color" value="#FFFF0000" />
    </text>
  </group>
</group>
```



# Panels and Animated Menus



## Animation Panels – Some Examples

- Animating a sliding panel that slides in and out of the parent frame
  - Popup that slides in from the bottom of the screen then slides back out when dismissed
  - Soft key menus
- Full screen menu panels
  - Large icon presentation
  - Emulate a scrolling behavior when moving from item to item

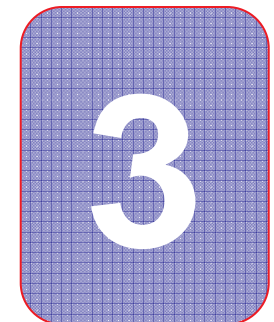
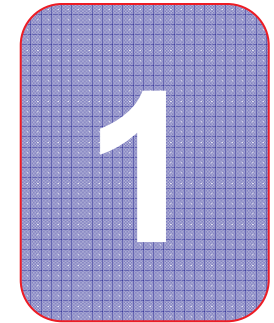


## Popup Panel

- Use a template approach to build a popup window
  - Build a single fragment for all popup windows
  - Use parameter loading to add the content
- Use `<slide>` element for entrance and exit

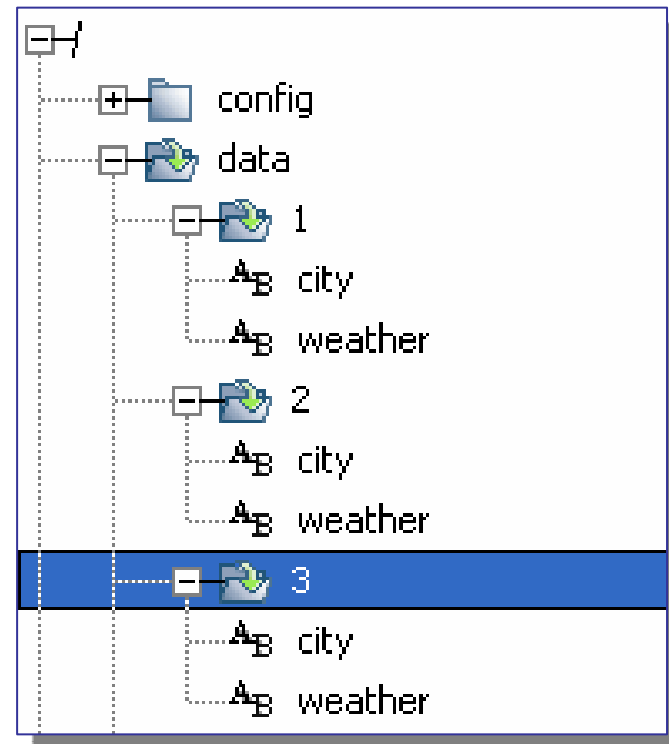
# Paneled Menu Example

- Build the panel data as resources that can be stepped through
  - Resources could be text, images, TrigML fragments, or a combination of all
- Build the layout components to display the appropriate panel
- Build the navigation sequence with animated transitions



# Construct the Data Panels

- Build a directory structure to hold the resources
  - One top level folder (data)
  - One folder for each panel
  - Numbered sequentially
- Add the resources to the directories
  - TrigML, images, text, or combination
  - Resources should be the same for each directory





## Build the Layout Components

- `<group>` element defines the window for the panel and clipping to hide the others
- Single column `<grid>` element for the menu
- Place a `<group>` element in each of the panel cells and point the repeatover to the data folder
- Build in event handling to navigate the menu
  - throw `moveup` and `movedown` events to your animation sequence



## Add the Transition Effects

- Build a <seq> element to process the moveup and movedown events thrown by the <grid>
- On movedown event
  - Use <anim> on the y coordinate to move <grid> up
  - Throw \_reverse to each group's repeatover to move backward in the data folder
- On moveup event
  - Use <anim> on the y coordinate to move <grid> down
  - Throw \_advance to each groups repeatover to move forward in the data folder



# Review

- Introduction
  - Some basics of animation sequences and TrigML
- Basic Animation Elements
  - Discuss tags that are used by TrigPlayer to render animation effects
- Animating Icons
  - Using animations on images to create interest in menus
- Creating Tickers
  - Using the basics learned above to create scrolling tickers and moving text
- Panels and Animated Menus
  - Setting up panels that move
- Q & A and Wrap-up



Questions?



## For More Information

- Check the BREW<sup>®</sup> Website
- Visit the uiOne<sup>™</sup> Pros downstairs in the Partner Pavilion
- Email [brew-training@qualcomm.com](mailto:brew-training@qualcomm.com) for all your training questions

Thanks and enjoy the  
conference!

