



# Capabilities Basics, Codecs etc. Creating Video Rules, tools & settings Integration Time for some code Interactivity Play with your video



# Flex Video: Capabilities



# Why Flash?

- 98% of Internet users in the US/Canada already have Flash Player 7 or higher.
- Flash Player 7 introduced expanded support for video. Flash Player 9 has even more.
- It allows easy integration with Flex and AIR applications.
- Flash is cross platform, even to many mobile devices.
- 70%+ of video streaming providers use Flash video

# Flex Video: Capabilities



# Why Flash Video?

- Ubiquity
- Expressiveness
- Customization/Control
- Quality
- Advanced Runtime Visual Effects & Capabilities
- Streaming capability
- Ease of implementation
- Already the standard for video delivery by sites like YouTube.

## Flex Video: Capabilities



# Codecs Overview: Sorenson

- Developed by Sorenson Media and licensed to Macromedia in 2002 for release with Flash MX.
- The current version is version 3, known as Sorenson Spark, and it can be used to create FLV files. Spark has Basic and Pro versions. Pro allows for higher image quality than the Basic version.
- Required for Flash 6 or 7 video deployment.
- Better for large frame sizes or video with high bit rates (over 1 Mbps), because it is less processor intensive.
- Good for targeting a wide range of devices, with varying processor power.

# Flex Video: Capabilities



# Codecs Overview: On2 VP6

- Developed by On2 Technologies and adopted by Macromedia for video in Flash Player 8.
- Produces high image quality, but is processor intensive. Comes in the normal and Simple flavors.
   Simple is basically uncompressed and can be HD, but is supported only in the latest Flash Player release.
- Good for target audiences with modern processors (Pentium IV and up) and Flash Player 8 and higher.
- Good for video smaller than 640 x 480.
- Supports alpha channels.

# Flex Video: Capabilities



- Codecs Overview: H.264 & AAC/AAC+
  - Developed by the MPEG and VCEG group as part of MPEG-4 and it includes high definition capability. It is used a lot in HD DVD/Blu Ray DVD.
  - Available in the most recent release of Flash Player 9 (Moviestar). Can only be streamed with FMS 3.
  - Supports true HD video, multi-core support, many devices, and lots of metadata.
  - Can be processor intensive, but hardware acceleration and in full screen mode allows for smooth playback of large video.
  - Note: moov atom issue



# Flex Video: Creating Video



- Creating (compression)
  - The rules
  - The tools
  - The settings

Kevin Towes -- http://www.adobe.com/devnet/flash/articles/flv\_encoding.html

# **Creating Video: Rules**



## **General Rules**

(frame height x frame width x frame rate (fps)) / Compression = total bits/sec.

- As the target bandwidth (data rate) decreases, reduce the keyframe rate.
- As motion increases, you must increase the keyframe rate, the frame rate, and the data rate.
- Dial-up connections consume almost all of the available bandwidth

# **Creating Video: Rules**



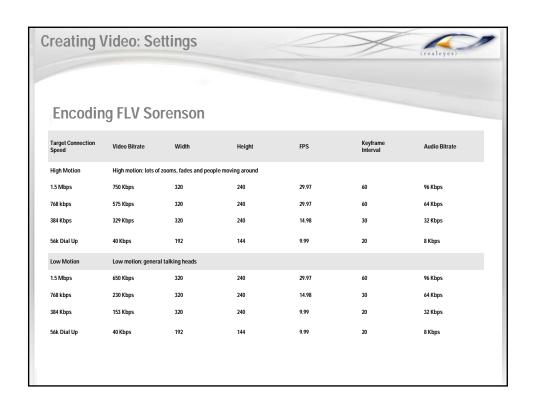
# **Rules Cont.**

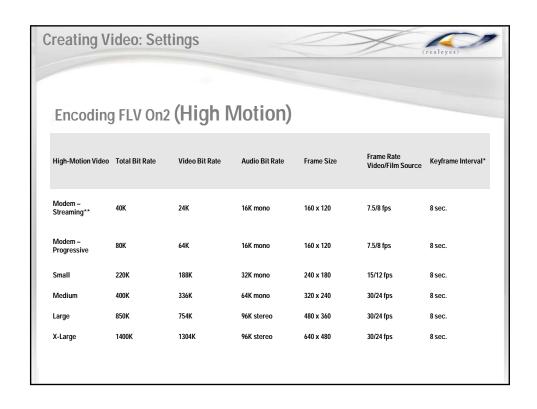
- Frame rates are calculated at half-rate, quarter-rate, and third-rate
- Reduce frame size when bandwidth is limited and frame rate and quality are important.
- Always enable de-interlacing and set the Flash Video encoder to the upper field.
- Extra video noise in lower quality video requires additional data rate.

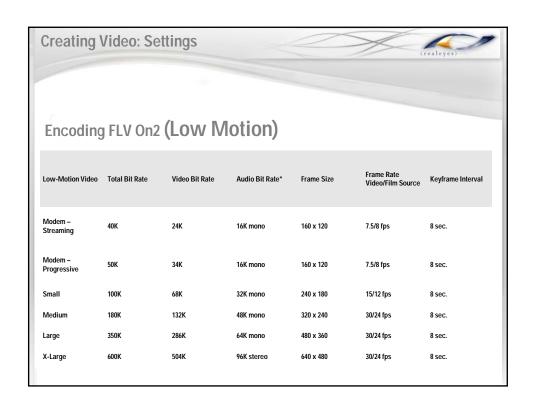
# **Creating Video: Tools**

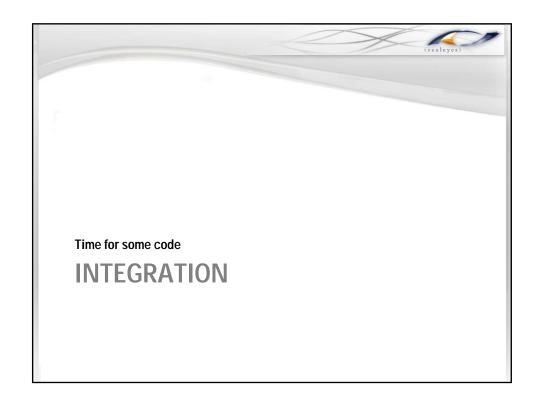


- Tools
  - RIVA (free)
  - FFMpeg (free)
  - On2Flix
  - Sorenson Squeeze
  - Creative Suite & Adobe Media Encoder









# Flex Video: Integration



- Integration
  - Component
  - Custom code
    - Connection > Stream: Video & Audio
    - Streaming H.264/AAC+ from FMS
  - Meta-data
  - Full-screen & hardware acceleration
  - Playback Media Player Style
  - Seamless display
    - Alpha
    - Layer Transform/Filter
    - Compositing

# **Integration: Flex Video Display**



# **Component: Flex Video Display**

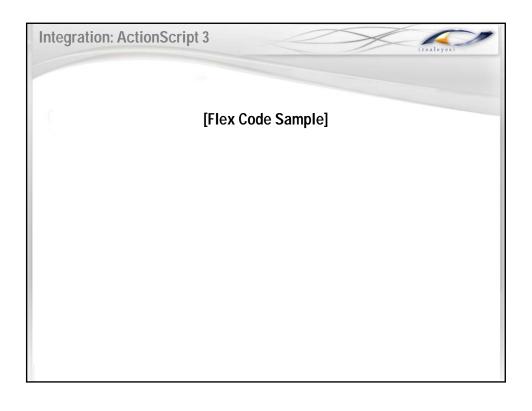
- No playback component for Flex, just the Video Display.
- Lots of methods and events exposed for control and cue point management.
- Use the VideoDisplay tag, specify a source, set dimensions to start.
- For deeper use, create a listener for video events to control playback.

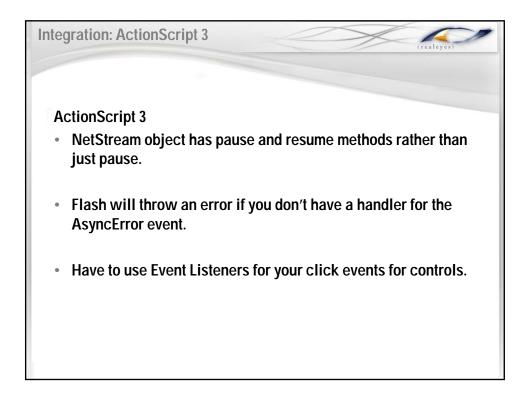
# **Integration: Flex Video Display**



- Using the existing/updated Flex 2.0 video display component
- The Flex 3.0 VideoDisplay component will play H.264 (.mov and .mp4 files, etc.) without any modifications.
- Using ActionScript 3.0 in Flex, you can create a custom class extending the UIComponent class that creates a Video object and uses the NetConnection and NetStream classes to play the video through the Video display object.

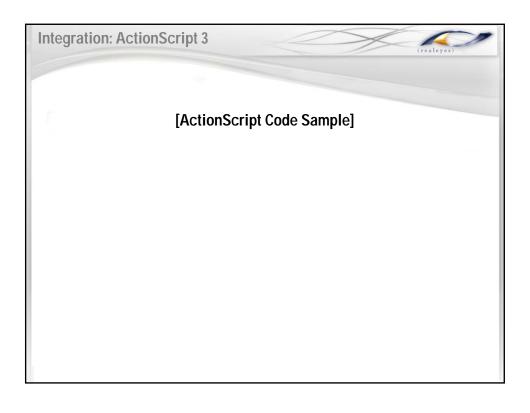
```
<mx:VideoDisplay id="reVid"</pre>
   width="320" height="240"
  autoBandWidthDetection="false|true"
  autoPlay="true|false"
  autoRewind="true|false"
  bufferTime="0.1"
  cuePointManagerClass=""
  cuePoints=""
  idleTimeout="300000"
  live="false|true"
  maintainAspectRatio="true|false"
  playheadTime=""
  playheadUpdateInterval="250"
  progressInterval="250"
  source="video/realeyes.flv"
  totalTime=""
  volume="0.75"/>
```

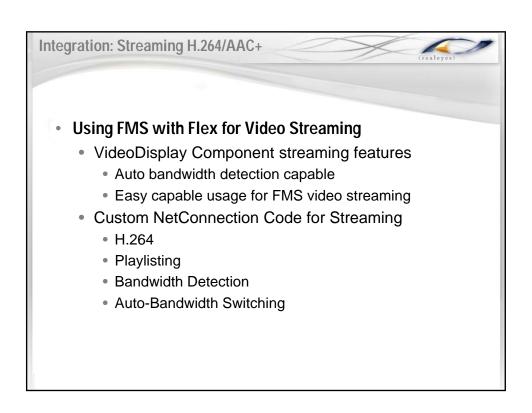




# Integration: Async Error Handler var video:Video = new Video(); //create the video display obj var connect\_nc:NetConnection = new NetConnection(); //create the net conn for external media connect\_nc.connect(null); //conn to null for progressive download connect\_nc.addEventListener(AsyncErrorEvent.ASYNC\_ERROR, errorHandler); var stream\_ns:NetStream = new NetStream(connect\_nc);//create a NetStream and hook it into the conn var \_client:Object = new Object(); //create a generic client object to receive all NetStream events

# 





Integration: Streaming H.264/AAC+



- Streaming H.264
  - Requires a streaming media server
  - Can stream or seek to any point non-linear
  - Quicker playback and much more secure
  - Flash Media Server will be the ONLY server to support H.264 and AAC+ streaming to Flash for now
  - Must use 'mp4:' prefix for stream play commands
  - Position of moov atom doesn't matter

stream\_ns.play( "mp4:fileToPlay.mp4" );

Integration: Streaming H.264/AAC+



- · Coding process stays similar as existing streaming.
- True streaming video does not have to have the moov atom at the beginning.
- FMS 3 supports encrypted streaming and enhanced seeking for MPEG-4 content, as well as all features currently supported for streaming FLV content.
- Capable of streaming HD quality video as well as live recording of HD quality video into H.264 format (via FME)!! (Now that's pretty slick)

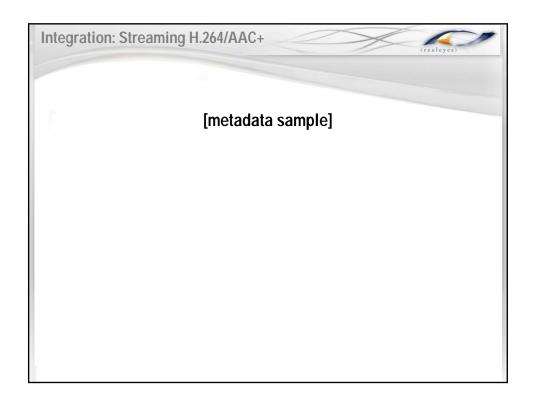
# H.264 Meta data • Using the onMetaData event of the NetStream object, you can extract information such as (depending on the encoder and codec): • Dimensions • Length • Codec

Other data that could potentially be stored in the meta data are:

• cover art

seek points

- subtitles
- · Audio book chapters
- \*If seek points aren't included in the meta data, you will be unable to seek in the file at all.
- \*Cue points currently are not supported.



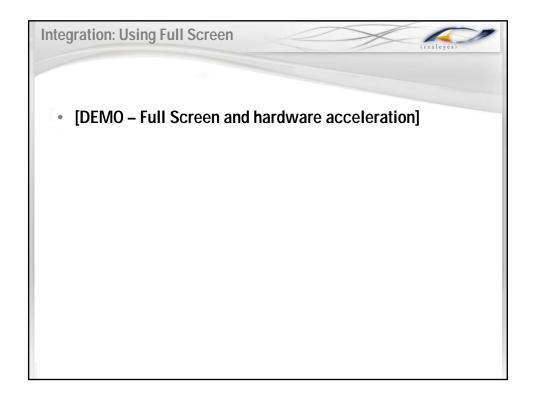
# **Integration: Using Full Screen**



- Full screen mode with hardware scaling
- The visual rendering process moves to the graphical processing unit (GPU).
- Hardware acceleration can be disabled in the new Display tab of the Flash Settings dialog box.
- API to specify a rectangular portion of the stage to be scaled:
  - Fewer pixels that need to be drawn to the screen
  - Better user experience
  - Lower CPU utilization for full-screen video playback.

# 

```
else
{
     stage.displayState = NORMAL;
}
catch(e:Error)
{
     Alert.show("Error Code: " + e.message, "FULL SCREEN ERROR");
}
}
```



# **Integration: Custom Playback Controls**

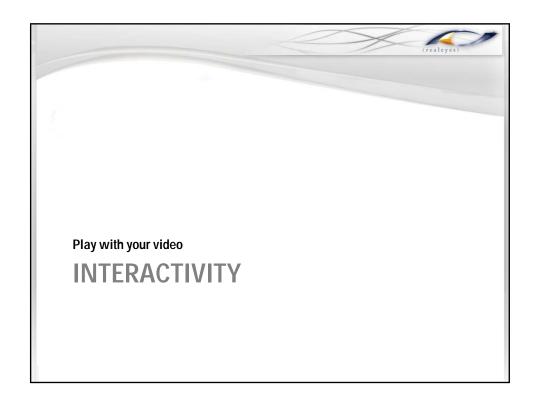


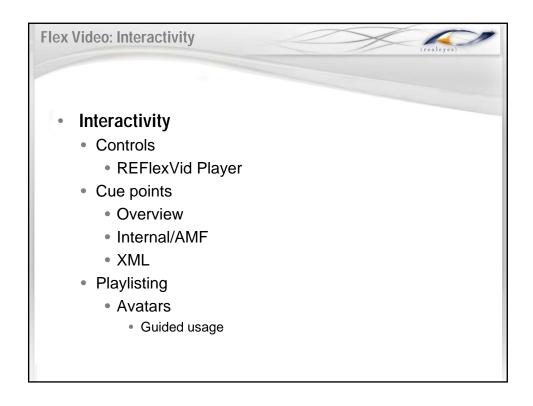
# Custom Playback Controls

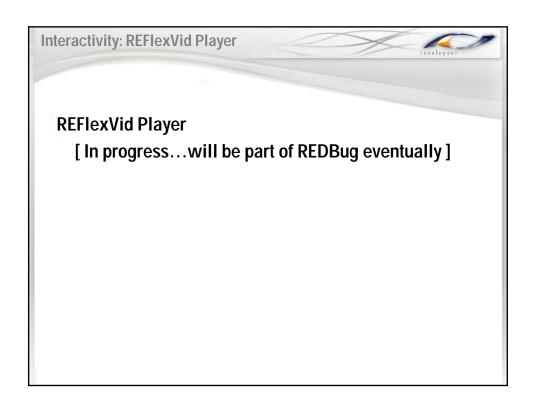
- For play/pause controls, use the NetStream's play and pause methods. For toggling play controls, you can check the class' togglePause method.
- The close method of the NetStream can act as stop.
- For fast forward and rewind, use the seek method.
- There is no playheadUpdate event equivalent for NetStream, so for scrub bars, you'll need to monitor playhead position (time property) through an enterFrame event or timer.
- Use the seek method for navigating with scrub bars.
- The HSlider component makes a handy scrub bar.

# Integration: Custom Playback Controls addEventListener(Event.ENTER\_FRAME, \_onEnterFrame, false, 0, true); private function \_onEnterFrame( evt:Event):void { position = \_stream\_ns.time; } public function get position():Number { return \_position; } public function set position(p\_value:Number):void { \_position = p\_value; \_scrubBar.value = \_position; \_stream\_ns.seek(\_position); }

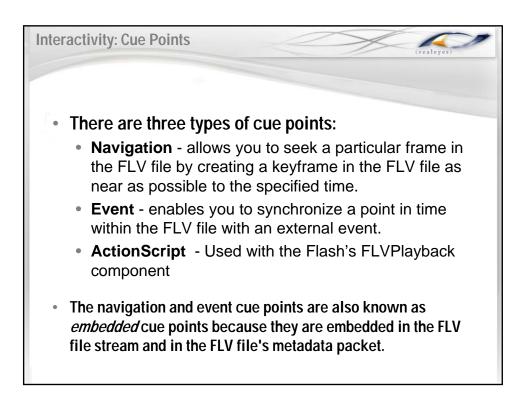
# Seamless display Alpha Video You can have an 8-bit alpha channel in videos encoded with the On2 VP6 codec SAMPLE: Combining H.264 & VP6 [Alpha VP6 video overlay on H.264]







# Cue Points Cue points Overview Internal/AMF XML



# **Interactivity: Cue Points**



- To handle the onCuePoint() call. The simplest solution is to set the NetStream client property equal to the this scope.
- Then the NetStream will look in the current scope for the onMetaData() and onCuePoint() methods.

```
var nc:NetConnection = new NetConnection();
nc.connect(null);
var ns:NetStream = new NetStream(nc);
ns.client = this;
...
function onCuePoint(infoObject:Object):void
{
    trace("cue point");
}
```

# Interactivity: REFlexVid Player



# **Playlisting**

- Avatars
  - Guided usage
  - XML Samples for the play list (progressive & w/ start/end)

```
Progressive sample

<data>
    <playlist>
        <video name="vid1.flv" title="Video 1" />
            <video name="vid2.flv" title="Video 2" />
            <video name="vid3.flv" title="Video 3" />
            <video name="vid4.flv" title="Video 4" />
            <video name="vid4.flv" title="Video 4" />
            <video name="vid5.flv" title="Video 5" />
            </playlist>
        </data>
```

# **Interactivity: Cue Points**



## TOP 10 TIPS - kinda:

- 1. Compression: Quality in, quality out
- 2. Use the right codec for the job
- 3. Be sure to handle your (asynch) events
- Video instances can NOT be added to containers use UIObject holders for light weight
- 5. Don't always try and build your application around the video. Use the creative side of the technology to integrate the video around your application.

# **Interactivity: Cue Points**



# • TOP 10 TIPS (continued):

- 6. Code reusability! Make/take a player and build upon it.
- 7. Fully explore all the classes associated with video and audio to see all the methods and properties available.
- 8. Write your mother. She loves you.
- 9. Meta data is your friend, respect it, talk to it, use it but in a nice way.
- 10. Know when you need to make a custom component and when a VideoDisplay component is more appropriate.

# Flex Video: Interactivity

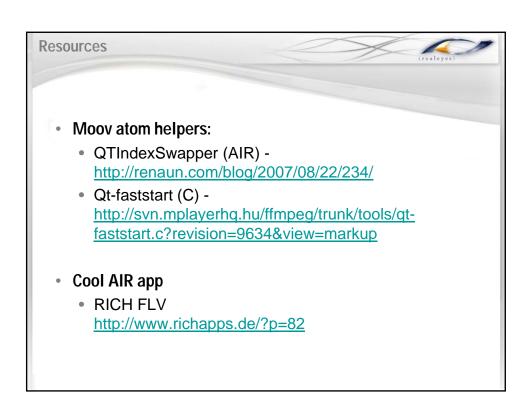


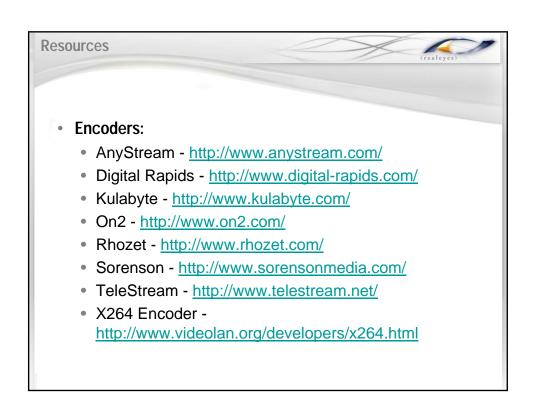
- Flash Video Dev Net -<a href="http://www.adobe.com/devnet/flash/video.html">http://www.adobe.com/devnet/flash/video.html</a>
- Flash Media Encoder -<a href="http://www.adobe.com/products/flashmediaserver/flashmediaencoder">http://www.adobe.com/products/flashmediaserver/flashmediaencoder</a>
- Timed Text XML Format http://www.w3.org/AudioVideo/TT/
- FLV Meta Data Injector <a href="http://www.asvguy.com/2005/06/flv\_metadata\_in.html">http://www.asvguy.com/2005/06/flv\_metadata\_in.html</a>

### Resources



- Adobe Labs: <a href="http://labs.adobe.com/">http://labs.adobe.com/</a>
- Moviestar Update: <a href="http://labs.adobe.com/technologies/flashplayer9/">http://labs.adobe.com/technologies/flashplayer9/</a>
- Tinic Uro's Blog: <a href="http://www.kaourantin.net/">http://www.kaourantin.net/</a>
- H.264 on WikiPedia: http://en.wikipedia.org/wiki/H264
- · AAC on WikiPedia: http://en.wikipedia.org/wiki/Advanced\_Audio\_Coding
- MPEG Industry Forum: www.mpeqif.org
- MPEG LA: www.mpeqla.com
- VIA Licensing: <u>www.vialicensing.com</u>
- Codec Comparisons:
  - <a href="http://www.streamingmedia.com/article.asp?id=9659&page=1&c=8">http://www.streamingmedia.com/article.asp?id=9659&page=1&c=8</a>
  - <a href="http://www.compression.ru/video/codec\_comparison/subjective\_codec\_comparison\_en.html">http://www.compression.ru/video/codec\_comparison/subjective\_codec\_comparison/subjective\_codec\_comparison/subjective\_codec\_comparison/subjective\_codec\_comparison/subjective\_codec\_comparison/subjective\_codec\_comparison/subjective\_codec\_comparison/subjective\_codec\_comparison/subjective\_codec\_comparison/subjective\_codec\_comparison/subjective\_codec\_comparison/subjective\_codec\_comparison/subjective\_codec\_comparison/subjective\_codec\_comparison/subjective\_codec\_comparison/subjective\_codec\_comparison/subjective\_codec\_comparison/subjective\_codec\_comparison\_codec\_codec\_comparison/subjective\_codec\_





# Resources



- David Hassoun david@realeyes.com
- Blog http://david.realeyes.com/
- Adobe Devnet Article
  <a href="http://www.adobe.com/devnet/flashplayer/articles/hd\_video\_flash\_player.html">http://www.adobe.com/devnet/flashplayer/articles/hd\_video\_flash\_player.html</a>
- Presentation & Additional Resources http://labs.realeyesmedia.com