转 ActionScript 3.0 API 中的 Video 类

2013年11月22日 00:10:45　阅读数：5760

注：这个类在Flash流媒体开发中使用的很频繁，在此记录一下它的使用方法。

包	flash.media
类	public class Video
继承	Video → DisplayObject → EventDispatcher → Object
子类	VideoPlayer

语言版本:	ActionScript 3.0

运行时版本:	AIR 1.0, Flash Player 9, Flash Lite 4

Video 类在应用程序中显示实时视频或录制视频，而无需在 SWF 文件中嵌入视频。此类创建播放下列任一类型视频的 Video 对象：在服务器上存储的或本地存储的录制的视频文件；用户捕获的实时视频。Video 对象是应用程序的显示列表中的一个显示对象，它表示在用户界面中运行视频的可见空间。
将 Video 对象与 Flash Media Server 一起使用时，可通过该对象将用户捕获的实时视频发送到服务器，然后从服务器向其他用户广播此视频。通过使用这些功能，您可以开发媒体应用程序，例如，简单的视频播放器、具有在服务器间进行多点发布功能的视频播放器或者用于用户社区的视频共享应用程序。

Flash Player 9 和更高版本支持发布和播放使用 Sorenson Spark 或 On2 VP6 编解码器编码的 FLV 文件，还支持 Alpha 通道。与以前的技术相比，On2 VP6 视频编解码器使用的带宽较少，并且提供了额外的马赛克消除滤镜和色度去环滤镜。有关视频播放和支持格式的详细信息，请参阅 flash.net.NetStream 类。

Flash Player 9.0.115.0 和更高版本支持 mipmap 处理以优化运行时，从而呈现品质和性能。对于视频播放，如果将 Video 对象的 smoothing 属性设置为 true，则 Flash Player 使用 mipmap 优化。

与显示列表中的其它显示对象一样，您可以控制 Video 对象的各种属性。例如，可以通过使用 Video 对象的 x 和 y 属性在舞台上移动该对象，以及通过使用其 height 和 width 属性更改其大小，等等。

要播放视频流，请使用 attachCamera() 或 attachNetStream() 将视频附加到 Video 对象。然后，使用 addChild() 将 Video 对象添加到显示列表中。

如果您使用的是 Flash Professional，则还可以将 Video 对象放在舞台上，而不是使用 addChild() 进行添加，如下所示：

1. 如果看不到“库”面板，请选择“窗口”>“库”，让该面板显示出来。
2. 单击“库”面板标题栏右侧的“选项”菜单，然后选择“新建视频”，在库中添加一个嵌入式 Video 对象。
3. 在“视频属性”对话框中，命名嵌入的 Video 对象以便在库中使用，然后单击“确定”。
4. 将该 Video 对象拖放到舞台上，然后使用属性检查器为其指定一个唯一的实例名称，如 my_video。（不要将其命名为“Video”。）

在桌面上的 AIR 应用程序中，以全屏模式播放视频会禁用任何电源和屏幕节省功能（如果操作系统允许）。

注意： Video 类不是 InteractiveObject 类的子类，因此它无法调度鼠标事件。但是，您可以对包含 Video 对象的显示对象容器调用 addEventListener() 方法。

公共属性

显示继承的公共属性

	属性	由以下参数定义
	deblocking : int 表示作为后处理的一部分应用于已解码视频的滤镜的类型。	Video
	smoothing : Boolean 指定在缩放视频时是否应进行平滑处理（插补数据）。	Video
	videoHeight : int [只读] 一个整数，以像素为单位指定视频流的高度。	Video

		videoWidth : int [只读] 一个整数，以像素为单位指定视频流的宽度。	Video

公共方法

显示继承的公共方法

	方法	由以下参数定义
	Video (width: int = 320, height: int = 240) 创建新的 Video 实例。	Video
	attachCamera (camera: Camera): void 指定要在应用程序中 Video 对象的边界内显示的来自摄像头的视频流。	Video
	attachNetStream (netStream: NetStream): void 指定要在应用程序中 Video 对象的边界内显示的视频流。	Video
	clear (): void 清除 Video 对象（而非视频流）中当前显示的图像。	Video

事件

单击此处了解有关事件的更多信息

显示继承的事件

属性详细信息

deblocking 属性

deblocking: int

语言版本:	ActionScript 3.0

运行时版本:	AIR 1.0, Flash Player 9, Flash Lite 4

表示作为后处理的一部分应用于已解码视频的滤镜的类型。默认值为 0，表示允许视频压缩程序根据需要应用消除马赛克的滤镜。

压缩视频可能会导致产生不希望有的伪像。可以使用 deblocking 属性设置用于减少马赛克（而对于使用 On2 编解码器压缩的视频，则用于降低扭曲边缘）的过滤器。

马赛克 是指组成每个视频帧的块的边界之间的可见瑕疵。 *扭曲边缘* 是指视频图像内元素周围的扭曲边缘。

有两种消除马赛克的滤镜：分别在 Sorenson 编解码器和 On2 VP6 编解码器中。此外，如果使用 On2 VP6 编解码器，则可以使用色度去环滤镜。要设置滤镜，请使用下列值之一：

- 0 — 允许视频压缩程序根据需要应用消除马赛克的滤镜。
- 1 — 不使用消除马赛克的滤镜。
- 2 — 使用 Sorenson 消除马赛克的滤镜。
- 3 —（只限 On2 视频）使用 On2 消除马赛克的滤镜，但不使用色度去环滤镜。
- 4 —（只限 On2 视频）使用 On2 消除马赛克的滤镜和色度去环滤镜。
- 5 —（只限 On2 视频）使用 On2 消除马赛克的滤镜和性能更高的 On2 色度去环滤镜。

如果在使用 Sorenson 编解码器时为视频选择了大于 2 的值，则 Sorenson 解码器默认为 2。

使用消除马赛克的滤镜会影响整体播放性能，而且对于高带宽视频通常没有必要。如果用户系统的功能不够强大，则用户在启用消除马赛克的滤镜的情况下播放视频可能会有困难。

实现

public function get deblocking(): int

public function set deblocking(value: int): void

smoothing 属性 | |

smoothing: Boolean

语言版本:	ActionScript 3.0

运行时版本:	AIR 1.0, Flash Player 9, Flash Lite 4

指定在缩放视频时是否应进行平滑处理(插补数据)。为顺畅工作,运行时必须处于高品质模式(默认值)。默认值是 `false`（不进行平滑处理）。

对于使用 Flash Player 9.0.115.0 及更高版本进行的视频播放，请将此属性设置为 `true` 以利用 mipmap 图像优化。

实现

public function get smoothing(): Boolean

public function set smoothing(value: Boolean): void

了解详细信息

利用 mipmap 处理

videoHeight	属性	

videoHeight: int [只读]

语言版本:	ActionScript 3.0

运行时版本:	AIR 1.0, Flash Player 9, Flash Lite 4

一个整数，以像素为单位指定视频流的高度。对于实时流，此值与正在捕获该视频流的 Camera 对象的 `Camera.height` 属性相同。对于录制的视频文件，此值为视频高度。对于录制的视频，当此值更改时，将会调度 NetStream.Video.DimensionChange 事件。

例如，您可能需要使用此属性来确保用户以捕获时的相同大小观看视频，而无论 Video 对象在舞台上的实际大小是什么。

实现

public function get videoHeight(): int

相关 API 元素

flash.media.Camera.height

videoWidth	属性	

videoWidth: int [只读]

语言版本:	ActionScript 3.0

运行时版本:	AIR 1.0, Flash Player 9, Flash Lite 4

一个整数,以像素为单位指定视频流的宽度。对于实时流,此值与正在捕获该视频流的 Camera 对象的 `Camera.width` 属性相同。对于录制的视频文件，此值为视频宽度。对于录制的视频，当此值更改时，将会调度 NetStream.Video.DimensionChange 事件。

例如，您可能需要使用此属性来确保用户以捕获时的相同大小观看视频，而无论 Video 对象在舞台上的实际大小是什么。

实现

public function get videoWidth(): int

相关 API 元素

flash.media.Camera.width

构造函数详细信息

Video	()	构造函数

public function Video(width: int = 320, height: int = 240)

语言版本:	ActionScript 3.0

运行时版本:	AIR 1.0, Flash Player 9

创建新的 Video 实例。如果没有提供 width 和 height 参数的值，将使用默认值。在初始构建后，也可以使用 Video.width 和 Video.height 设置 Video 对象的宽度和高度属性。在创建新的 Video 对象时，不允许宽度和高度的值为零；如果传递零，则将应用默认值。

创建 Video 后，请调用 DisplayObjectContainer.addChild() 或 DisplayObjectContainer.addChildAt() 方法以便将 Video 对象添加到父级 DisplayObjectContainer 对象。

参数

width : int (default = 320) — 视频的宽度（以像素为单位）。
height : int (default = 240) — 视频的高度，以像素为单位。

示例（如何使用本示例）

以下示例显示如何加载外部 FLV 文件：

```
var MyVideo:Video = new Video();
addChild(MyVideo);

var MyNC:NetConnection = new NetConnection();
MyNC.connect(null);

var MyNS:NetStream = new NetStream(MyNC);
MyNS.play("http://www.helpexamples.com/flash/video/clouds.flv");

MyVideo.attachNetStream(MyNS);

//the clouds.flv video has metadata we're not using, so create
//an error handler to ignore the message generated by the runtime
//about the metadata
MyNS.addEventListener(AsyncErrorEvent.ASYNC_ERROR, asyncErrorHandler);

function asyncErrorHandler(event:AsyncErrorEvent):void
{
  //ignore metadata error message
}
```

方法详细信息

attachCamera	()	方法

public function attachCamera(camera: Camera): void

语言版本:	ActionScript 3.0

运行时版本:	AIR 1.0, Flash Player 9

指定要在应用程序中 Video 对象的边界内显示的来自摄像头的视频流。

使用此方法将用户捕获的实时视频附加到 Video 对象。您可以在捕获实时视频的同一台计算机或设备上本地播放该视频，也可以将其发送到 Flash Media Server，然后使用该服务器将视频流式传输到其他用户。

注意： 在 iOS AIR 应用程序中，当应用程序使用 GPU 呈现模式时，不能显示摄像头视频。

参数

camera : Camera — 正在捕获视频数据的 Camera 对象。要切断与该 Video 对象的连接，请传递 null 。

相关 API 元素

Video.attachNetStream()
flash.media.Camera

示例
如何使用本示例
有关如何使用此方法的说明，请参阅 Camera.getCamera() 方法示例。

attachNetStream	()	方法	

public function attachNetStream(netStream: NetStream): void

语言版本:	ActionScript 3.0

运行时版本:	AIR 1.0, Flash Player 9, Flash Lite 4

指定要在应用程序中 Video 对象的边界内显示的视频流。视频流是使用 NetStream.play() 播放的视频文件，Camera 对象或 null。如果使用视频文件，则可以将其存储在本地文件系统或 Flash Media Server 中。如果 netStream 参数的值为 null，则不会再在 Video 对象中播放视频。

如果视频文件只包含音频，则无需使用此方法；当调用 NetStream.play() 时，将自动播放视频文件的音频部分。要控制与视频文件相关联的音频，请使用用于播放视频文件的 NetStream 对象的 soundTransform 属性。

参数

	netStream : NetStream — 一个 NetStream 对象。要切断与该 Video 对象的连接，请传递 null。

相关 API 元素

Video.attachCamera()
flash.net.NetStream.soundTransform
flash.net.NetStream.play()
SoundTransform

clear	()	方法	

public function clear(): void

语言版本:	ActionScript 3.0

运行时版本:	AIR 1.0, Flash Player 9, Flash Lite 4

清除 Video 对象（而非视频流）中当前显示的图像。此方法对于处理当前图像非常有用。例如，您可以清除最后一幅图像或显示待机信息，而无需隐藏 Video 对象。

相关 API 元素

Video.attachCamera()

示例 **如何使用本示例**

VideoExample.as

以下示例将 Video 对象与 NetConnection 和 NetStream 类一同使用，以加载和播放 FLV 文件。为了运行此示例，需要一个名称和位置与传递给 videoURL 的变量相匹配的 FLV 文件，在本例中，使用 SWF 文件所在目录下的一个名为 Video.flv 的 FLV 文件。
在本例中，创建 Video 和 NetStream 对象并调用 Video.attachNetStream() 和 NetStream.play() 的代码放入处理函数。仅在试图连接 NetConnection 对象成功时调用处理函数，即，当 netStatus 事件返回一个 info 对象，该对象带有一个表示成功的 code 属性时，才调用该处理函数。建议先等待连接成功，然后再调用 NetStream.play()。

```
package {
  import flash.display.Sprite;
  import flash.events.*;
  import flash.media.Video;
  import flash.net.NetConnection;
  import flash.net.NetStream;

  public class VideoExample extends Sprite {
    private var videoURL:String = "Video.flv";
    private var connection:NetConnection;
    private var stream:NetStream;

    public function VideoExample() {
      connection = new NetConnection();
      connection.addEventListener(NetStatusEvent.NET_STATUS, netStatusHandler);
      connection.addEventListener(SecurityErrorEvent.SECURITY_ERROR, securityErrorHandler);
      connection.connect(null);
    }

    private function netStatusHandler(event:NetStatusEvent):void {
      switch (event.info.code) {
        case "NetConnection.Connect.Success":
          connectStream();
          break;
        case "NetStream.Play.StreamNotFound":
          trace("Unable to locate video: " + videoURL);
          break;
      }
    }

    private function connectStream():void {
      stream = new NetStream(connection);
      stream.addEventListener(NetStatusEvent.NET_STATUS, netStatusHandler);
      stream.addEventListener(AsyncErrorEvent.ASYNC_ERROR, asyncErrorHandler);
      var video:Video = new Video();
      video.attachNetStream(stream);
      stream.play(videoURL);
      addChild(video);
    }

    private function securityErrorHandler(event:SecurityErrorEvent):void {
      trace("securityErrorHandler: " + event);
    }

    private function asyncErrorHandler(event:AsyncErrorEvent):void {
      // ignore AsyncErrorEvent events.
    }
  }
}
```

原文地址： http://help.adobe.com/zh_CN/FlashPlatform/reference/actionscript/3/flash/media/Video.html

文章标签： actionscript video 视频 播放 api

个人分类： Flash相关