<?xml version="1.0" encoding="utf-8"?>

<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"

xmlns:s="library://ns.adobe.com/flex/spark"

xmlns:mx="library://ns.adobe.com/flex/mx"

minWidth="2000" minHeight="800"

initialize="InitFlashPlayer()"

mouseMove="application\_mouseMoveHandler(event)" >

<!-- FlashPlayerDemo VideoDisplay-->

<!-- local video, http request for record, rtmp request for live -->

<fx:Script>

<![CDATA[

import flash.display.StageDisplayState;

import flash.events.MouseEvent;

import flash.events.TimerEvent;

import flash.utils.Timer;

import mx.controls.Text;

import mx.core.FlexGlobals;

import mx.events.FlexEvent;

import mx.events.SliderEvent;

import mx.events.VideoEvent;

import org.osmf.events.LoadEvent;

import org.osmf.events.TimeEvent;

import spark.components.Button;

import spark.events.TrackBaseEvent;

[Embed(source="Source/Resource/play.png")] //Embed the resource

[Bindable]

private var playImage:Class; //bind the variable

[Embed(source="Source/Resource/pause.png")]

[Bindable]

private var pauseImage:Class;

private var isFullScreen:Boolean = false; // 默认不是全屏

private var videoSourceURL:String;

private var arr = new Array();

protected function InitFlashPlayer():void

{

hslider.track.alpha = 0;

showtime.text = "00:00/00:00";

border.setStyle("showEffect",fade);

videoSourceURL = "Source/VideoSource/study.mp4";

//videoSourceURL = "http://blog.minidx.com/ext/getting-started-with-the-videodisplay-control/clouds.flv";

//videoSourceURL = "rtmp://192.241.59.220:1935/live/045C0609A529BPL\_0\_2?code=38669535244";

videodisplay.source = videoSourceURL;

}

protected function videodisplay\_completeHandler(event:TimeEvent):void

{

play.label = "暂停";

this.play.setStyle("icon",this.pauseImage);

videodisplay.play();

}

protected function play\_clickHandler(event:MouseEvent):void

{

if(play.label== "播放")

{

play.label = "暂停";

canvas.visible = false;

this.play.setStyle("icon",this.pauseImage);

videodisplay.play();

trace("播放" + videodisplay.currentTime);

}

else if(play.label == "暂停")

{

play.label = "播放";

canvas.visible = true;

this.play.setStyle("icon",this.playImage);

videodisplay.pause();

trace("暂停" + videodisplay.currentTime);

}

}

protected function stop\_clickHandler(event:MouseEvent):void

{

videodisplay.stop();

play.label = "播放";

canvas.visible = true;

this.play.setStyle("icon",this.playImage);

}

protected function volume\_changeHandler(event:SliderEvent):void

{

videodisplay.volume = volume.value/100; //音量

}

protected function hslider\_mouseMoveHandler(event:MouseEvent):void

{

currenttime.visible = true;

currenttime.x = hslider.x+hslider.mouseX;

currenttime.text = parseTime(progress.mouseX/progress.width\*videodisplay.duration);

}

protected function hslider\_mouseOutHandler(event:MouseEvent):void

{

currenttime.visible = false;

}

protected function hslider\_changeHandler(event:Event):void

{

videodisplay.seek(hslider.value/100\*videodisplay.duration);

if(play.label == "播放")

{

play.label = "暂停";

canvas.visible=false;

this.play.setStyle("icon",this.pauseImage);

}

}

protected function hslider\_thumbPressHandler(event:TrackBaseEvent):void

{

videodisplay.pause();

}

protected function hslider\_thumbReleaseHandler(event:TrackBaseEvent):void

{

videodisplay.play();

}

protected function videodisplay\_doubleClickHandler(event:MouseEvent):void

{

fullscreen\_clickHandler(event);

}

protected function videodisplay\_currentTimeChangeHandler(event:TimeEvent):void

{

showtime.text = parseTime(videodisplay.currentTime) + "/" + parseTime(videodisplay.duration);

hslider.value = videodisplay.currentTime/videodisplay.duration\*100;

}

protected function videodisplay\_bytesLoadedChangeHandler(event:LoadEvent):void

{

progress.setProgress(videodisplay.bytesLoaded,videodisplay.bytesTotal);

}

protected function hslider\_thumbDragHandler(event:TrackBaseEvent):void

{

videodisplay.seek(hslider.value);

}

protected function fullscreen\_clickHandler(event:MouseEvent):void

{

if(!isFullScreen) //enterFullScreen

{

stage.displayState=StageDisplayState.FULL\_SCREEN;

stage.addEventListener(FullScreenEvent.FULL\_SCREEN, fullScreenEventHandler);

videodisplay.width = stage.width/1.2;

videodisplay.height = stage.height/1.2;

full.width = stage.width/1.2;

full.height = stage.height/1.2;

canvas.width = stage.width/1.2;

canvas.height = stage.height/1.2;

borderD.y=stage.height-25;

borderD.width=stage.width/1.2-1;

border.width=stage.width/1.2-1;

fullscreen.x=stage.width/1.2-25;

volume.x=stage.width-75;

voice.x=stage.width-90;

showtime.x=stage.width-155;

progress.x=330;

hslider.x=330;

progress.width=stage.width/1.2-205;

hslider.width=stage.width/1.2-205;

border.visible = true;

currenttime.visible = false;

isFullScreen = true;

}

else //exitFullscreen

{

stage.displayState = StageDisplayState.NORMAL;

videodisplay.width = 334;

videodisplay.height = 285;

full.width = 334;

full.height = 285;

canvas.width = 334;

canvas.height = 285;

borderD.y=259;

borderD.width=333;

border.width=333;

progress.width=160;

hslider.width=160;

fullscreen.width=16;

fullscreen.x=313;

showtime.x=192;

volume.x=264;

voice.x=254;

currenttime.y=-3;

isFullScreen = false;

}

}

public function fullScreenEventHandler(evn:FullScreenEvent):void

{

stage.removeEventListener(FullScreenEvent.FULL\_SCREEN, fullScreenEventHandler);

play.visible = true;

stop.visible = true;

volume.visible=true;

fullscreen.visible = true;

showtime.visible = true;

border.visible = true;

progress.visible = true;

hslider.visible = true;

videodisplay.width = 334;

videodisplay.height = 285;

full.width = 334;

full.height = 285;

canvas.width = 334;

canvas.height = 285;

borderD.y=259;

borderD.width=333;

border.width=333;

progress.width=160;

hslider.width=160;

fullscreen.width=16;

fullscreen.x=313;

showtime.x=192;

volume.x=264;

voice.x=254;

currenttime.y=-3;

isFullScreen = false;

}

protected function parseTime(Time:int):String

{

var minute:int;

var second:int;

minute = Time/60;

second = Time - (minute\*60);

var d:String = new String();

if(minute < 10 && second < 10)

{

d = "0" + minute.toString(10)+":"+ "0" + second.toString(10);

}

else if(minute < 10 && second >= 10)

{

d = "0" + minute.toString(10)+":"+ second.toString(10);

}

else if(minute >= 10 && second < 10)

{

d = minute.toString(10)+":"+ "0" + second.toString(10);

}

else if(minute >= 10 && second >= 10)

{

d = minute.toString(10)+":"+ second.toString(10);

}

return d;

}

protected function application\_mouseMoveHandler(event:MouseEvent):void

{

if(videodisplay.width==334)

{

if(event.localX > 0 && event.localX < 334 && event.stageY > 0 && event.stageY < 285)

{

border.visible = true;

return ;

}

else

{

border.visible = false;

return ;

}

}

else

{

if(event.stageY > stage.height-20 && event.stageY <= stage.height)

{

border.visible = true;

return ;

}

else

{

border.visible = false;

return ;

}

}

}

]]>

</fx:Script>

<fx:Declarations>

<!-- 将非可视元素（例如服务、值对象）放在此处 -->

<mx:Fade id="fade" />

<mx:Zoom id="zoom" />

</fx:Declarations>

<!-- 视频大小 -->

<s:VideoDisplay id="videodisplay" x="0" y="0" width="2020" height="1000" autoPlay="true"

currentTimeChange="videodisplay\_currentTimeChangeHandler(event)"

autoDisplayFirstFrame="true" autoRewind="true" bytesLoadedChange="videodisplay\_bytesLoadedChangeHandler(event)"

complete="videodisplay\_completeHandler(event)" opaqueBackground="#000000"

buttonMode="true" useHandCursor="true"/>

<!--全屏-->

<mx:Label id="full" x="0" y="0" height="1000" width="2020" doubleClickEnabled="true"

doubleClick="videodisplay\_doubleClickHandler(event)" alpha="0"/>

<!--暂停-->

<mx:Canvas id="canvas" x="0" y="0" width="2020" height="285" borderSkin="@Embed(source='Source/Resource/pause.swf')" visible="false"

doubleClickEnabled="true" doubleClick="fullscreen\_clickHandler(event)"

click="play\_clickHandler(event)"/>

<s:BorderContainer id="borderD" x="0" y="500" width="2020" height="100" backgroundColor="#000000" borderAlpha="0"

backgroundAlpha="0.0">

<!--下拉框-->

<s:BorderContainer id="border" x="0" y="500" width="2020" height="100" borderColor="#FE0000"

backgroundColor="#030003" visible="false" backgroundAlpha="0.5" borderAlpha="0" >

<!--停止logo-->

<mx:LinkButton id="play" x="5" y="25" label="暂停" height="50" width="50" fontFamily="微软雅黑" toggle="true"

click="play\_clickHandler(event)" alpha="1" toolTip="播放/暂停" fontSize="16"

icon="@Embed(source='Source/Resource/pause.png')"/>

<!--暂停logo-->

<mx:LinkButton id="stop" x="50" y="25" label="停止" height="50" width="50" fontFamily="微软雅黑"

click="stop\_clickHandler(event)" toolTip="停止" fontSize="10"

icon="@Embed(source='Source/Resource/stop.png')" alpha="1" toggle="true"/>

<!--全屏logo-->

<mx:LinkButton id="fullscreen" x="1955" y="25" label="全屏" height="50" width="50"

fontFamily="微软雅黑" click="fullscreen\_clickHandler(event)" toolTip="全屏" fontSize="10"

icon="@Embed(source='Source/Resource/fullscreen.png')"/>

<!--时间显示-->

<s:TextInput id="showtime" x="1600" y="30" width="200" contentBackgroundColor="#000000" fontSize="30" editable="false"

fontFamily="仿宋" color="#FFFFFF" contentBackgroundAlpha="0.0" borderAlpha="1" borderVisible="false" height="50"/>

<!--进度条-->

<mx:ProgressBar x="100" y="45" fontSize="0" labelWidth="0" height="30" width="1500" chromeColor="#000000"

barSkin="mx.skins.halo.ProgressBarSkin" trackHeight="6" minimum="0" maximum="100" mode="manual"

id="progress" fontFamily="微软雅黑" useHandCursor="true"/>

<!--滚动条-->

<s:HSlider id="hslider" x="110" y="35" width="1500" height="30" minimum="0" maximum="100"

showDataTip="false" mouseMove="hslider\_mouseMoveHandler(event)" mouseOut="hslider\_mouseOutHandler(event)"

change="hslider\_changeHandler(event)" thumbPress="hslider\_thumbPressHandler(event)" liveDragging="true"

thumbRelease="hslider\_thumbReleaseHandler(event)" useHandCursor="true" buttonMode="true" skinClass="myhslider"

thumbDrag="hslider\_thumbDragHandler(event)"/>

<!--音量进度条-->

<mx:HSlider id="volume" visible="true" x="1855" y="25" width="80" height="50"

change="volume\_changeHandler(event)" enabled="true"

maximum="100" minimum="0" showDataTip="true" showTrackHighlight="true"

value="50"/>

<!--音量logo-->

<mx:Image id="voice" x="1805" y="33" source="Source/Resource/speed.png" width="45" height="45" toolTip="速度"/>

<s:Label id="currenttime" x="33" y="-3" width="75" height="17" visible="true" fontSize="10" color="#FFFFFF"

fontWeight="bold" backgroundColor="#000000" backgroundAlpha="0.0"/>

</s:BorderContainer>

</s:BorderContainer>

</s:Application>

<?xml version="1.0" encoding="utf-8"?>

<s:SparkSkin xmlns:fx="http://ns.adobe.com/mxml/2009"

xmlns:s="library://ns.adobe.com/flex/spark"

xmlns:fb="http://ns.adobe.com/flashbuilder/2009"

minHeight="11" alpha.disabled="0.5">

<fx:Metadata>

<![CDATA[

/\*\*

\* @copy spark.skins.spark.ApplicationSkin#hostComponent

\*/

[HostComponent("spark.components.HSlider")]

]]>

</fx:Metadata>

<!--

NOTE: this skin file contains sub-parts that may continue to react to

Style code. To remove this behavior create new copies of those skins

and remove the styles.

-->

<fx:Script>

/\*\*

\* @private

\*/

override protected function measure() : void

{

// Temporarily move the thumb to the left of the Slider so measurement

// doesn't factor in its x position. This allows resizing the

// HSlider to less than 100px in width.

var thumbPos:Number = thumb.getLayoutBoundsX();

thumb.setLayoutBoundsPosition(0, thumb.getLayoutBoundsY());

super.measure();

thumb.setLayoutBoundsPosition(thumbPos, thumb.getLayoutBoundsY());

}

</fx:Script>

<s:states>

<s:State name="normal" />

<s:State name="disabled" />

</s:states>

<fx:Declarations>

<!--- The tooltip used in the mx.controls.Slider control.

To customize the DataTip's appearance, create a custom HSliderSkin class.-->

<fx:Component id="dataTip">

<s:DataRenderer minHeight="24" minWidth="40" y="-34">

<s:Rect top="0" left="0" right="0" bottom="0">

<s:fill>

<s:SolidColor color="0x123456" alpha=".9"/>

</s:fill>

<s:filters>

<s:DropShadowFilter angle="90" color="0x999999" distance="3"/>

</s:filters>

</s:Rect>

<s:Label id="labelDisplay" text="{data}"

horizontalCenter="0" verticalCenter="1"

left="5" right="5" top="5" bottom="5"

textAlign="center" verticalAlign="middle"

fontWeight="normal" color="white" fontSize="11">

</s:Label>

</s:DataRenderer>

</fx:Component>

</fx:Declarations>

<!--- The default skin class is HSliderTrackSkin.

@copy spark.components.supportClasses.TrackBase#track

@see spark.skins.spark.HSliderTrackSkin -->

<s:Button id="track" left="0" right="0" top="0" bottom="0" minWidth="33" width="100"

skinClass="spark.skins.spark.HSliderTrackSkin" />

<!--- The default skin class is HSliderThumbSkin.

@copy spark.components.supportClasses.TrackBase#thumb

@see spark.skins.spark.HSliderThumbSkin -->

<s:Button id="thumb" top="0" bottom="0" width="18" height="11"

skinClass="spark.skins.spark.ButtonSkin" height.normal="5" width.normal="10"/>

</s:SparkSkin>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<!-- saved from url=(0014)about:internet -->

<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">

<!--

Smart developers always View Source.

This application was built using Adobe Flex, an open source framework

for building rich Internet applications that get delivered via the

Flash Player or to desktops via Adobe AIR.

Learn more about Flex at http://flex.org

// -->

<head>

<title></title>

<meta name="google" value="notranslate" />

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<style type="text/css" media="screen">

html, body { height:1100px}

body { margin:10px; padding:10px; overflow:auto; text-align:center;

background-color: pink; }

object:focus { outline:none; }

#flashContent { display:none; }

</style>

<!-- Enable Browser History by replacing useBrowserHistory tokens with two hyphens -->

<!-- BEGIN Browser History required section -->

<link rel="stylesheet" type="text/css" href="history/history.css" />

<script type="text/javascript" src="history/history.js"></script>

<!-- END Browser History required section -->

<script type="text/javascript" src="swfobject.js"></script>

<script type="text/javascript">

var swfVersionStr = "10.1.0";

var xiSwfUrlStr = "playerProductInstall.swf";

var flashvars = {};

var params = {};

params.quality = "high";

params.bgcolor = "#ffffff";

params.allowscriptaccess = "sameDomain";

params.allowfullscreen = "true";

var attributes = {};

attributes.id = "FlashPlayerDemo";

attributes.name = "FlashPlayerDemo";

attributes.align = "middle";

swfobject.embedSWF(

"FlashPlayerDemo.swf", "flashContent",

"100%", "100%",

swfVersionStr, xiSwfUrlStr,

flashvars, params, attributes);

// JavaScript enabled so display the flashContent div in case it is not replaced with a swf object.

swfobject.createCSS("#flashContent", "display:block;text-align:left;");

</script>

</head>

<body style="background-color:grey">

<!-- SWFObject's dynamic embed method replaces this alternative HTML content with Flash content when enough

JavaScript and Flash plug-in support is available. The div is initially hidden so that it doesn't show

when JavaScript is disabled.

-->

<div style="font-family: 'kaiti'; font-size:40px">

<p>

请输入视频网址：<input type="text" name="" value="http://" style="width:50%;border:1px solid grey">

<button type="button" name="button">播放</button>

</p>

</div>

<div id="flashContent">

<p>

To view this page ensure that Adobe Flash Player version

11.1.0 or greater is installed.

</p>

<script type="text/javascript">

var pageHost = ((document.location.protocol == "https:") ? "https://" : "http://");

document.write("<a href='http://www.adobe.com/go/getflashplayer'><img src='"

+ pageHost + "www.adobe.com/images/shared/download\_buttons/get\_flash\_player.gif' alt='Get Adobe Flash player' /></a>" );

</script>

</div>

<noscript>

<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" width="100%" height="600px" id="FlashPlayerDemo">

<param name="movie" value="FlashPlayerDemo.swf" />

<param name="quality" value="high" />

<param name="bgcolor" value="#ffffff" />

<param name="allowScriptAccess" value="sameDomain" />

<param name="allowFullScreen" value="true" />

<!--[if !IE]>-->

<object type="application/x-shockwave-flash" data="FlashPlayerDemo.swf" width="100%" height="80%">

<param name="quality" value="high" />

<param name="bgcolor" value="#ffffff" />

<param name="allowScriptAccess" value="sameDomain" />

<param name="allowFullScreen" value="true" />

<!--<![endif]-->

<!--[if gte IE 6]>-->

<p>

Either scripts and active content are not permitted to run or Adobe Flash Player version

11.1.0 or greater is not installed.

</p>

<!--<![endif]-->

<a href="http://www.adobe.com/go/getflashplayer">

<img src="http://www.adobe.com/images/shared/download\_buttons/get\_flash\_player.gif" alt="Get Adobe Flash Player" />

</a>

<!--[if !IE]>-->

</object>

<!--<![endif]-->

</object>

</noscript>

</body>

</html>

/\* This CSS stylesheet defines styles used by required elements in a flex application page that supports browser history \*/

#ie\_historyFrame { width: 0px; height: 0px; display:none }

#firefox\_anchorDiv { width: 0px; height: 0px; display:none }

#safari\_formDiv { width: 0px; height: 0px; display:none }

#safari\_rememberDiv { width: 0px; height: 0px; display:none }

BrowserHistoryUtils = {

addEvent: function(elm, evType, fn, useCapture) {

useCapture = useCapture || false;

if (elm.addEventListener) {

elm.addEventListener(evType, fn, useCapture);

return true;

}

else if (elm.attachEvent) {

var r = elm.attachEvent('on' + evType, fn);

return r;

}

else {

elm['on' + evType] = fn;

}

}

}

BrowserHistory = (function() {

// type of browser

var browser = {

ie: false,

ie8: false,

firefox: false,

safari: false,

opera: false,

version: -1

};

// Default app state URL to use when no fragment ID present

var defaultHash = '';

// Last-known app state URL

var currentHref = document.location.href;

// Initial URL (used only by IE)

var initialHref = document.location.href;

// Initial URL (used only by IE)

var initialHash = document.location.hash;

// History frame source URL prefix (used only by IE)

var historyFrameSourcePrefix = 'history/historyFrame.html?';

// History maintenance (used only by Safari)

var currentHistoryLength = -1;

// Flag to denote the existence of onhashchange

var browserHasHashChange = false;

var historyHash = [];

var initialState = createState(initialHref, initialHref + '#' + initialHash, initialHash);

var backStack = [];

var forwardStack = [];

var currentObjectId = null;

//UserAgent detection

var useragent = navigator.userAgent.toLowerCase();

if (useragent.indexOf("opera") != -1) {

browser.opera = true;

} else if (useragent.indexOf("msie") != -1) {

browser.ie = true;

browser.version = parseFloat(useragent.substring(useragent.indexOf('msie') + 4));

if (browser.version == 8)

{

browser.ie = false;

browser.ie8 = true;

}

} else if (useragent.indexOf("safari") != -1) {

browser.safari = true;

browser.version = parseFloat(useragent.substring(useragent.indexOf('safari') + 7));

} else if (useragent.indexOf("gecko") != -1) {

browser.firefox = true;

}

if (browser.ie == true && browser.version == 7) {

window["\_ie\_firstload"] = false;

}

function hashChangeHandler()

{

currentHref = document.location.href;

var flexAppUrl = getHash();

//ADR: to fix multiple

if (typeof BrowserHistory\_multiple != "undefined" && BrowserHistory\_multiple == true) {

var pl = getPlayers();

for (var i = 0; i < pl.length; i++) {

pl[i].browserURLChange(flexAppUrl);

}

} else {

getPlayer().browserURLChange(flexAppUrl);

}

}

// Accessor functions for obtaining specific elements of the page.

function getHistoryFrame()

{

return document.getElementById('ie\_historyFrame');

}

function getFormElement()

{

return document.getElementById('safari\_formDiv');

}

function getRememberElement()

{

return document.getElementById("safari\_remember\_field");

}

// Get the Flash player object for performing ExternalInterface callbacks.

// Updated for changes to SWFObject2.

function getPlayer(id) {

var i;

if (id && document.getElementById(id)) {

var r = document.getElementById(id);

if (typeof r.SetVariable != "undefined") {

return r;

}

else {

var o = r.getElementsByTagName("object");

var e = r.getElementsByTagName("embed");

for (i = 0; i < o.length; i++) {

if (typeof o[i].browserURLChange != "undefined")

return o[i];

}

for (i = 0; i < e.length; i++) {

if (typeof e[i].browserURLChange != "undefined")

return e[i];

}

}

}

else {

var o = document.getElementsByTagName("object");

var e = document.getElementsByTagName("embed");

for (i = 0; i < e.length; i++) {

if (typeof e[i].browserURLChange != "undefined")

{

return e[i];

}

}

for (i = 0; i < o.length; i++) {

if (typeof o[i].browserURLChange != "undefined")

{

return o[i];

}

}

}

return undefined;

}

function getPlayers() {

var i;

var players = [];

if (players.length == 0) {

var tmp = document.getElementsByTagName('object');

for (i = 0; i < tmp.length; i++)

{

if (typeof tmp[i].browserURLChange != "undefined")

players.push(tmp[i]);

}

}

if (players.length == 0 || players[0].object == null) {

var tmp = document.getElementsByTagName('embed');

for (i = 0; i < tmp.length; i++)

{

if (typeof tmp[i].browserURLChange != "undefined")

players.push(tmp[i]);

}

}

return players;

}

function getIframeHash() {

var doc = getHistoryFrame().contentWindow.document;

var hash = String(doc.location.search);

if (hash.length == 1 && hash.charAt(0) == "?") {

hash = "";

}

else if (hash.length >= 2 && hash.charAt(0) == "?") {

hash = hash.substring(1);

}

return hash;

}

/\* Get the current location hash excluding the '#' symbol. \*/

function getHash() {

// It would be nice if we could use document.location.hash here,

// but it's faulty sometimes.

var idx = document.location.href.indexOf('#');

return (idx >= 0) ? document.location.href.substr(idx+1) : '';

}

/\* Get the current location hash excluding the '#' symbol. \*/

function setHash(hash) {

// It would be nice if we could use document.location.hash here,

// but it's faulty sometimes.

if (hash == '') hash = '#'

document.location.hash = hash;

}

function createState(baseUrl, newUrl, flexAppUrl) {

return { 'baseUrl': baseUrl, 'newUrl': newUrl, 'flexAppUrl': flexAppUrl, 'title': null };

}

function addHistoryEntry(baseUrl, newUrl, flexAppUrl) {

//delete all the history entries

forwardStack = [];

if (browser.ie) {

//Check to see if we are being asked to do a navigate for the first

//history entry, and if so ignore, because it's coming from the creation

//of the history iframe

if (flexAppUrl == defaultHash && document.location.href == initialHref && window['\_ie\_firstload']) {

currentHref = initialHref;

return;

}

if ((!flexAppUrl || flexAppUrl == defaultHash) && window['\_ie\_firstload']) {

newUrl = baseUrl + '#' + defaultHash;

flexAppUrl = defaultHash;

} else {

// for IE, tell the history frame to go somewhere without a '#'

// in order to get this entry into the browser history.

getHistoryFrame().src = historyFrameSourcePrefix + flexAppUrl;

}

setHash(flexAppUrl);

} else {

//ADR

if (backStack.length == 0 && initialState.flexAppUrl == flexAppUrl) {

initialState = createState(baseUrl, newUrl, flexAppUrl);

} else if(backStack.length > 0 && backStack[backStack.length - 1].flexAppUrl == flexAppUrl) {

backStack[backStack.length - 1] = createState(baseUrl, newUrl, flexAppUrl);

}

if (browser.safari && !browserHasHashChange) {

// for Safari, submit a form whose action points to the desired URL

if (browser.version <= 419.3) {

var file = window.location.pathname.toString();

file = file.substring(file.lastIndexOf("/")+1);

getFormElement().innerHTML = '<form name="historyForm" action="'+file+'#' + flexAppUrl + '" method="GET"></form>';

//get the current elements and add them to the form

var qs = window.location.search.substring(1);

var qs\_arr = qs.split("&");

for (var i = 0; i < qs\_arr.length; i++) {

var tmp = qs\_arr[i].split("=");

var elem = document.createElement("input");

elem.type = "hidden";

elem.name = tmp[0];

elem.value = tmp[1];

document.forms.historyForm.appendChild(elem);

}

document.forms.historyForm.submit();

} else {

top.location.hash = flexAppUrl;

}

// We also have to maintain the history by hand for Safari

historyHash[history.length] = flexAppUrl;

\_storeStates();

} else {

// Otherwise, just tell the browser to go there

setHash(flexAppUrl);

}

}

backStack.push(createState(baseUrl, newUrl, flexAppUrl));

}

function \_storeStates() {

if (browser.safari) {

getRememberElement().value = historyHash.join(",");

}

}

function handleBackButton() {

//The "current" page is always at the top of the history stack.

var current = backStack.pop();

if (!current) { return; }

var last = backStack[backStack.length - 1];

if (!last && backStack.length == 0){

last = initialState;

}

forwardStack.push(current);

}

function handleForwardButton() {

//summary: private method. Do not call this directly.

var last = forwardStack.pop();

if (!last) { return; }

backStack.push(last);

}

function handleArbitraryUrl() {

//delete all the history entries

forwardStack = [];

}

function checkForUrlChange() {

if (browser.ie) {

if (currentHref != document.location.href && currentHref + '#' != document.location.href) {

if (browser.version < 7) {

currentHref = document.location.href;

document.location.reload();

} else {

if (getHash() != getIframeHash()) {

// this.iframe.src = this.blankURL + hash;

var sourceToSet = historyFrameSourcePrefix + getHash();

getHistoryFrame().src = sourceToSet;

currentHref = document.location.href;

}

}

}

}

if (browser.safari && !browserHasHashChange) {

// For Safari, we have to check to see if history.length changed.

if (currentHistoryLength >= 0 && history.length != currentHistoryLength) {

//alert("did change: " + history.length + ", " + historyHash.length + "|" + historyHash[history.length] + "|>" + historyHash.join("|"));

var flexAppUrl = getHash();

if (browser.version < 528.16 /\* Anything earlier than Safari 4.0 \*/)

{

// If it did change and we're running Safari 3.x or earlier,

// then we have to look the old state up in our hand-maintained

// array since document.location.hash won't have changed,

// then call back into BrowserManager.

currentHistoryLength = history.length;

flexAppUrl = historyHash[currentHistoryLength];

}

//ADR: to fix multiple

if (typeof BrowserHistory\_multiple != "undefined" && BrowserHistory\_multiple == true) {

var pl = getPlayers();

for (var i = 0; i < pl.length; i++) {

pl[i].browserURLChange(flexAppUrl);

}

} else {

getPlayer().browserURLChange(flexAppUrl);

}

\_storeStates();

}

}

if (browser.firefox && !browserHasHashChange) {

if (currentHref != document.location.href) {

var bsl = backStack.length;

var urlActions = {

back: false,

forward: false,

set: false

}

if ((window.location.hash == initialHash || window.location.href == initialHref) && (bsl == 1)) {

urlActions.back = true;

// FIXME: could this ever be a forward button?

// we can't clear it because we still need to check for forwards. Ugg.

// clearInterval(this.locationTimer);

handleBackButton();

}

// first check to see if we could have gone forward. We always halt on

// a no-hash item.

if (forwardStack.length > 0) {

if (forwardStack[forwardStack.length-1].flexAppUrl == getHash()) {

urlActions.forward = true;

handleForwardButton();

}

}

// ok, that didn't work, try someplace back in the history stack

if ((bsl >= 2) && (backStack[bsl - 2])) {

if (backStack[bsl - 2].flexAppUrl == getHash()) {

urlActions.back = true;

handleBackButton();

}

}

if (!urlActions.back && !urlActions.forward) {

var foundInStacks = {

back: -1,

forward: -1

}

for (var i = 0; i < backStack.length; i++) {

if (backStack[i].flexAppUrl == getHash() && i != (bsl - 2)) {

arbitraryUrl = true;

foundInStacks.back = i;

}

}

for (var i = 0; i < forwardStack.length; i++) {

if (forwardStack[i].flexAppUrl == getHash() && i != (bsl - 2)) {

arbitraryUrl = true;

foundInStacks.forward = i;

}

}

handleArbitraryUrl();

}

// Firefox changed; do a callback into BrowserManager to tell it.

currentHref = document.location.href;

var flexAppUrl = getHash();

//ADR: to fix multiple

if (typeof BrowserHistory\_multiple != "undefined" && BrowserHistory\_multiple == true) {

var pl = getPlayers();

for (var i = 0; i < pl.length; i++) {

pl[i].browserURLChange(flexAppUrl);

}

} else {

getPlayer().browserURLChange(flexAppUrl);

}

}

}

}

var \_initialize = function () {

browserHasHashChange = ("onhashchange" in document.body);

if (browser.ie)

{

var scripts = document.getElementsByTagName('script');

for (var i = 0, s; s = scripts[i]; i++) {

if (s.src.indexOf("history.js") > -1) {

var iframe\_location = (new String(s.src)).replace("history.js", "historyFrame.html");

}

}

historyFrameSourcePrefix = iframe\_location + "?";

var src = historyFrameSourcePrefix;

var iframe = document.createElement("iframe");

iframe.id = 'ie\_historyFrame';

iframe.name = 'ie\_historyFrame';

iframe.src = 'javascript:false;';

try {

document.body.appendChild(iframe);

} catch(e) {

setTimeout(function() {

document.body.appendChild(iframe);

}, 0);

}

}

if (browser.safari && !browserHasHashChange)

{

var rememberDiv = document.createElement("div");

rememberDiv.id = 'safari\_rememberDiv';

document.body.appendChild(rememberDiv);

rememberDiv.innerHTML = '<input type="text" id="safari\_remember\_field" style="width: 500px;">';

var formDiv = document.createElement("div");

formDiv.id = 'safari\_formDiv';

document.body.appendChild(formDiv);

var reloader\_content = document.createElement('div');

reloader\_content.id = 'safarireloader';

var scripts = document.getElementsByTagName('script');

for (var i = 0, s; s = scripts[i]; i++) {

if (s.src.indexOf("history.js") > -1) {

html = (new String(s.src)).replace(".js", ".html");

}

}

reloader\_content.innerHTML = '<iframe id="safarireloader-iframe" src="about:blank" frameborder="no" scrolling="no"></iframe>';

document.body.appendChild(reloader\_content);

reloader\_content.style.position = 'absolute';

reloader\_content.style.left = reloader\_content.style.top = '-9999px';

iframe = reloader\_content.getElementsByTagName('iframe')[0];

if (document.getElementById("safari\_remember\_field").value != "" ) {

historyHash = document.getElementById("safari\_remember\_field").value.split(",");

}

}

if (browserHasHashChange)

document.body.onhashchange = hashChangeHandler;

}

return {

historyHash: historyHash,

backStack: function() { return backStack; },

forwardStack: function() { return forwardStack },

getPlayer: getPlayer,

initialize: function(src) {

\_initialize(src);

},

setURL: function(url) {

document.location.href = url;

},

getURL: function() {

return document.location.href;

},

getTitle: function() {

return document.title;

},

setTitle: function(title) {

try {

backStack[backStack.length - 1].title = title;

} catch(e) { }

//if on safari, set the title to be the empty string.

if (browser.safari) {

if (title == "") {

try {

var tmp = window.location.href.toString();

title = tmp.substring((tmp.lastIndexOf("/")+1), tmp.lastIndexOf("#"));

} catch(e) {

title = "";

}

}

}

document.title = title;

},

setDefaultURL: function(def)

{

defaultHash = def;

def = getHash();

if (browser.ie)

{

window['\_ie\_firstload'] = true;

var sourceToSet = historyFrameSourcePrefix + def;

var func = function() {

getHistoryFrame().src = sourceToSet;

window.location.replace("#" + def);

setInterval(checkForUrlChange, 50);

}

try {

func();

} catch(e) {

window.setTimeout(function() { func(); }, 0);

}

}

if (browser.safari)

{

currentHistoryLength = history.length;

if (historyHash.length == 0) {

historyHash[currentHistoryLength] = def;

var newloc = "#" + def;

window.location.replace(newloc);

} else {

//alert(historyHash[historyHash.length-1]);

}

setInterval(checkForUrlChange, 50);

}

if (browser.firefox || browser.opera)

{

var reg = new RegExp("#" + def + "$");

if (window.location.toString().match(reg)) {

} else {

var newloc ="#" + def;

window.location.replace(newloc);

}

setInterval(checkForUrlChange, 50);

}

},

/\* Set the current browser URL; called from inside BrowserManager to propagate

\* the application state out to the container.

\*/

setBrowserURL: function(flexAppUrl, objectId) {

if (browser.ie && typeof objectId != "undefined") {

currentObjectId = objectId;

}

//fromIframe = fromIframe || false;

//fromFlex = fromFlex || false;

//alert("setBrowserURL: " + flexAppUrl);

//flexAppUrl = (flexAppUrl == "") ? defaultHash : flexAppUrl ;

var pos = document.location.href.indexOf('#');

var baseUrl = pos != -1 ? document.location.href.substr(0, pos) : document.location.href;

var newUrl = baseUrl + '#' + flexAppUrl;

if (document.location.href != newUrl && document.location.href + '#' != newUrl) {

currentHref = newUrl;

addHistoryEntry(baseUrl, newUrl, flexAppUrl);

currentHistoryLength = history.length;

}

},

browserURLChange: function(flexAppUrl) {

var objectId = null;

if (browser.ie && currentObjectId != null) {

objectId = currentObjectId;

}

if (typeof BrowserHistory\_multiple != "undefined" && BrowserHistory\_multiple == true) {

var pl = getPlayers();

for (var i = 0; i < pl.length; i++) {

try {

pl[i].browserURLChange(flexAppUrl);

} catch(e) { }

}

} else {

try {

getPlayer(objectId).browserURLChange(flexAppUrl);

} catch(e) { }

}

currentObjectId = null;

},

getUserAgent: function() {

return navigator.userAgent;

},

getPlatform: function() {

return navigator.platform;

}

}

})();

// Initialization

// Automated unit testing and other diagnostics

function setURL(url)

{

document.location.href = url;

}

function backButton()

{

history.back();

}

function forwardButton()

{

history.forward();

}

function goForwardOrBackInHistory(step)

{

history.go(step);

}

//BrowserHistoryUtils.addEvent(window, "load", function() { BrowserHistory.initialize(); });

(function(i) {

var u =navigator.userAgent;var e=/\*@cc\_on!@\*/false;

var st = setTimeout;

if(/webkit/i.test(u)){

st(function(){

var dr=document.readyState;

if(dr=="loaded"||dr=="complete"){i()}

else{st(arguments.callee,10);}},10);

} else if((/mozilla/i.test(u)&&!/(compati)/.test(u)) || (/opera/i.test(u))){

document.addEventListener("DOMContentLoaded",i,false);

} else if(e){

(function(){

var t=document.createElement('doc:rdy');

try{t.doScroll('left');

i();t=null;

}catch(e){st(arguments.callee,0);}})();

} else{

window.onload=i;

}

})( function() {BrowserHistory.initialize();} );

/\*! SWFObject v2.2 <http://code.google.com/p/swfobject/>

is released under the MIT License <http://www.opensource.org/licenses/mit-license.php>

\*/

var swfobject = function() {

var UNDEF = "undefined",

OBJECT = "object",

SHOCKWAVE\_FLASH = "Shockwave Flash",

SHOCKWAVE\_FLASH\_AX = "ShockwaveFlash.ShockwaveFlash",

FLASH\_MIME\_TYPE = "application/x-shockwave-flash",

EXPRESS\_INSTALL\_ID = "SWFObjectExprInst",

ON\_READY\_STATE\_CHANGE = "onreadystatechange",

win = window,

doc = document,

nav = navigator,

plugin = false,

domLoadFnArr = [main],

regObjArr = [],

objIdArr = [],

listenersArr = [],

storedAltContent,

storedAltContentId,

storedCallbackFn,

storedCallbackObj,

isDomLoaded = false,

isExpressInstallActive = false,

dynamicStylesheet,

dynamicStylesheetMedia,

autoHideShow = true,

/\* Centralized function for browser feature detection

- User agent string detection is only used when no good alternative is possible

- Is executed directly for optimal performance

\*/

ua = function() {

var w3cdom = typeof doc.getElementById != UNDEF && typeof doc.getElementsByTagName != UNDEF && typeof doc.createElement != UNDEF,

u = nav.userAgent.toLowerCase(),

p = nav.platform.toLowerCase(),

windows = p ? /win/.test(p) : /win/.test(u),

mac = p ? /mac/.test(p) : /mac/.test(u),

webkit = /webkit/.test(u) ? parseFloat(u.replace(/^.\*webkit\/(\d+(\.\d+)?).\*$/, "$1")) : false, // returns either the webkit version or false if not webkit

ie = !+"\v1", // feature detection based on Andrea Giammarchi's solution: http://webreflection.blogspot.com/2009/01/32-bytes-to-know-if-your-browser-is-ie.html

playerVersion = [0,0,0],

d = null;

if (typeof nav.plugins != UNDEF && typeof nav.plugins[SHOCKWAVE\_FLASH] == OBJECT) {

d = nav.plugins[SHOCKWAVE\_FLASH].description;

if (d && !(typeof nav.mimeTypes != UNDEF && nav.mimeTypes[FLASH\_MIME\_TYPE] && !nav.mimeTypes[FLASH\_MIME\_TYPE].enabledPlugin)) { // navigator.mimeTypes["application/x-shockwave-flash"].enabledPlugin indicates whether plug-ins are enabled or disabled in Safari 3+

plugin = true;

ie = false; // cascaded feature detection for Internet Explorer

d = d.replace(/^.\*\s+(\S+\s+\S+$)/, "$1");

playerVersion[0] = parseInt(d.replace(/^(.\*)\..\*$/, "$1"), 10);

playerVersion[1] = parseInt(d.replace(/^.\*\.(.\*)\s.\*$/, "$1"), 10);

playerVersion[2] = /[a-zA-Z]/.test(d) ? parseInt(d.replace(/^.\*[a-zA-Z]+(.\*)$/, "$1"), 10) : 0;

}

}

else if (typeof win.ActiveXObject != UNDEF) {

try {

var a = new ActiveXObject(SHOCKWAVE\_FLASH\_AX);

if (a) { // a will return null when ActiveX is disabled

d = a.GetVariable("$version");

if (d) {

ie = true; // cascaded feature detection for Internet Explorer

d = d.split(" ")[1].split(",");

playerVersion = [parseInt(d[0], 10), parseInt(d[1], 10), parseInt(d[2], 10)];

}

}

}

catch(e) {}

}

return { w3:w3cdom, pv:playerVersion, wk:webkit, ie:ie, win:windows, mac:mac };

}(),

/\* Cross-browser onDomLoad

- Will fire an event as soon as the DOM of a web page is loaded

- Internet Explorer workaround based on Diego Perini's solution: http://javascript.nwbox.com/IEContentLoaded/

- Regular onload serves as fallback

\*/

onDomLoad = function() {

if (!ua.w3) { return; }

if ((typeof doc.readyState != UNDEF && doc.readyState == "complete") || (typeof doc.readyState == UNDEF && (doc.getElementsByTagName("body")[0] || doc.body))) { // function is fired after onload, e.g. when script is inserted dynamically

callDomLoadFunctions();

}

if (!isDomLoaded) {

if (typeof doc.addEventListener != UNDEF) {

doc.addEventListener("DOMContentLoaded", callDomLoadFunctions, false);

}

if (ua.ie && ua.win) {

doc.attachEvent(ON\_READY\_STATE\_CHANGE, function() {

if (doc.readyState == "complete") {

doc.detachEvent(ON\_READY\_STATE\_CHANGE, arguments.callee);

callDomLoadFunctions();

}

});

if (win == top) { // if not inside an iframe

(function(){

if (isDomLoaded) { return; }

try {

doc.documentElement.doScroll("left");

}

catch(e) {

setTimeout(arguments.callee, 0);

return;

}

callDomLoadFunctions();

})();

}

}

if (ua.wk) {

(function(){

if (isDomLoaded) { return; }

if (!/loaded|complete/.test(doc.readyState)) {

setTimeout(arguments.callee, 0);

return;

}

callDomLoadFunctions();

})();

}

addLoadEvent(callDomLoadFunctions);

}

}();

function callDomLoadFunctions() {

if (isDomLoaded) { return; }

try { // test if we can really add/remove elements to/from the DOM; we don't want to fire it too early

var t = doc.getElementsByTagName("body")[0].appendChild(createElement("span"));

t.parentNode.removeChild(t);

}

catch (e) { return; }

isDomLoaded = true;

var dl = domLoadFnArr.length;

for (var i = 0; i < dl; i++) {

domLoadFnArr[i]();

}

}

function addDomLoadEvent(fn) {

if (isDomLoaded) {

fn();

}

else {

domLoadFnArr[domLoadFnArr.length] = fn; // Array.push() is only available in IE5.5+

}

}

/\* Cross-browser onload

- Based on James Edwards' solution: http://brothercake.com/site/resources/scripts/onload/

- Will fire an event as soon as a web page including all of its assets are loaded

\*/

function addLoadEvent(fn) {

if (typeof win.addEventListener != UNDEF) {

win.addEventListener("load", fn, false);

}

else if (typeof doc.addEventListener != UNDEF) {

doc.addEventListener("load", fn, false);

}

else if (typeof win.attachEvent != UNDEF) {

addListener(win, "onload", fn);

}

else if (typeof win.onload == "function") {

var fnOld = win.onload;

win.onload = function() {

fnOld();

fn();

};

}

else {

win.onload = fn;

}

}

/\* Main function

- Will preferably execute onDomLoad, otherwise onload (as a fallback)

\*/

function main() {

if (plugin) {

testPlayerVersion();

}

else {

matchVersions();

}

}

/\* Detect the Flash Player version for non-Internet Explorer browsers

- Detecting the plug-in version via the object element is more precise than using the plugins collection item's description:

a. Both release and build numbers can be detected

b. Avoid wrong descriptions by corrupt installers provided by Adobe

c. Avoid wrong descriptions by multiple Flash Player entries in the plugin Array, caused by incorrect browser imports

- Disadvantage of this method is that it depends on the availability of the DOM, while the plugins collection is immediately available

\*/

function testPlayerVersion() {

var b = doc.getElementsByTagName("body")[0];

var o = createElement(OBJECT);

o.setAttribute("type", FLASH\_MIME\_TYPE);

var t = b.appendChild(o);

if (t) {

var counter = 0;

(function(){

if (typeof t.GetVariable != UNDEF) {

var d = t.GetVariable("$version");

if (d) {

d = d.split(" ")[1].split(",");

ua.pv = [parseInt(d[0], 10), parseInt(d[1], 10), parseInt(d[2], 10)];

}

}

else if (counter < 10) {

counter++;

setTimeout(arguments.callee, 10);

return;

}

b.removeChild(o);

t = null;

matchVersions();

})();

}

else {

matchVersions();

}

}

/\* Perform Flash Player and SWF version matching; static publishing only

\*/

function matchVersions() {

var rl = regObjArr.length;

if (rl > 0) {

for (var i = 0; i < rl; i++) { // for each registered object element

var id = regObjArr[i].id;

var cb = regObjArr[i].callbackFn;

var cbObj = {success:false, id:id};

if (ua.pv[0] > 0) {

var obj = getElementById(id);

if (obj) {

if (hasPlayerVersion(regObjArr[i].swfVersion) && !(ua.wk && ua.wk < 312)) { // Flash Player version >= published SWF version: Houston, we have a match!

setVisibility(id, true);

if (cb) {

cbObj.success = true;

cbObj.ref = getObjectById(id);

cb(cbObj);

}

}

else if (regObjArr[i].expressInstall && canExpressInstall()) { // show the Adobe Express Install dialog if set by the web page author and if supported

var att = {};

att.data = regObjArr[i].expressInstall;

att.width = obj.getAttribute("width") || "0";

att.height = obj.getAttribute("height") || "0";

if (obj.getAttribute("class")) { att.styleclass = obj.getAttribute("class"); }

if (obj.getAttribute("align")) { att.align = obj.getAttribute("align"); }

// parse HTML object param element's name-value pairs

var par = {};

var p = obj.getElementsByTagName("param");

var pl = p.length;

for (var j = 0; j < pl; j++) {

if (p[j].getAttribute("name").toLowerCase() != "movie") {

par[p[j].getAttribute("name")] = p[j].getAttribute("value");

}

}

showExpressInstall(att, par, id, cb);

}

else { // Flash Player and SWF version mismatch or an older Webkit engine that ignores the HTML object element's nested param elements: display alternative content instead of SWF

displayAltContent(obj);

if (cb) { cb(cbObj); }

}

}

}

else { // if no Flash Player is installed or the fp version cannot be detected we let the HTML object element do its job (either show a SWF or alternative content)

setVisibility(id, true);

if (cb) {

var o = getObjectById(id); // test whether there is an HTML object element or not

if (o && typeof o.SetVariable != UNDEF) {

cbObj.success = true;

cbObj.ref = o;

}

cb(cbObj);

}

}

}

}

}

function getObjectById(objectIdStr) {

var r = null;

var o = getElementById(objectIdStr);

if (o && o.nodeName == "OBJECT") {

if (typeof o.SetVariable != UNDEF) {

r = o;

}

else {

var n = o.getElementsByTagName(OBJECT)[0];

if (n) {

r = n;

}

}

}

return r;

}

/\* Requirements for Adobe Express Install

- only one instance can be active at a time

- fp 6.0.65 or higher

- Win/Mac OS only

- no Webkit engines older than version 312

\*/

function canExpressInstall() {

return !isExpressInstallActive && hasPlayerVersion("6.0.65") && (ua.win || ua.mac) && !(ua.wk && ua.wk < 312);

}

/\* Show the Adobe Express Install dialog

- Reference: http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=6a253b75

\*/

function showExpressInstall(att, par, replaceElemIdStr, callbackFn) {

isExpressInstallActive = true;

storedCallbackFn = callbackFn || null;

storedCallbackObj = {success:false, id:replaceElemIdStr};

var obj = getElementById(replaceElemIdStr);

if (obj) {

if (obj.nodeName == "OBJECT") { // static publishing

storedAltContent = abstractAltContent(obj);

storedAltContentId = null;

}

else { // dynamic publishing

storedAltContent = obj;

storedAltContentId = replaceElemIdStr;

}

att.id = EXPRESS\_INSTALL\_ID;

if (typeof att.width == UNDEF || (!/%$/.test(att.width) && parseInt(att.width, 10) < 310)) { att.width = "310"; }

if (typeof att.height == UNDEF || (!/%$/.test(att.height) && parseInt(att.height, 10) < 137)) { att.height = "137"; }

doc.title = doc.title.slice(0, 47) + " - Flash Player Installation";

var pt = ua.ie && ua.win ? "ActiveX" : "PlugIn",

fv = "MMredirectURL=" + encodeURI(window.location).toString().replace(/&/g,"%26") + "&MMplayerType=" + pt + "&MMdoctitle=" + doc.title;

if (typeof par.flashvars != UNDEF) {

par.flashvars += "&" + fv;

}

else {

par.flashvars = fv;

}

// IE only: when a SWF is loading (AND: not available in cache) wait for the readyState of the object element to become 4 before removing it,

// because you cannot properly cancel a loading SWF file without breaking browser load references, also obj.onreadystatechange doesn't work

if (ua.ie && ua.win && obj.readyState != 4) {

var newObj = createElement("div");

replaceElemIdStr += "SWFObjectNew";

newObj.setAttribute("id", replaceElemIdStr);

obj.parentNode.insertBefore(newObj, obj); // insert placeholder div that will be replaced by the object element that loads expressinstall.swf

obj.style.display = "none";

(function(){

if (obj.readyState == 4) {

obj.parentNode.removeChild(obj);

}

else {

setTimeout(arguments.callee, 10);

}

})();

}

createSWF(att, par, replaceElemIdStr);

}

}

/\* Functions to abstract and display alternative content

\*/

function displayAltContent(obj) {

if (ua.ie && ua.win && obj.readyState != 4) {

// IE only: when a SWF is loading (AND: not available in cache) wait for the readyState of the object element to become 4 before removing it,

// because you cannot properly cancel a loading SWF file without breaking browser load references, also obj.onreadystatechange doesn't work

var el = createElement("div");

obj.parentNode.insertBefore(el, obj); // insert placeholder div that will be replaced by the alternative content

el.parentNode.replaceChild(abstractAltContent(obj), el);

obj.style.display = "none";

(function(){

if (obj.readyState == 4) {

obj.parentNode.removeChild(obj);

}

else {

setTimeout(arguments.callee, 10);

}

})();

}

else {

obj.parentNode.replaceChild(abstractAltContent(obj), obj);

}

}

function abstractAltContent(obj) {

var ac = createElement("div");

if (ua.win && ua.ie) {

ac.innerHTML = obj.innerHTML;

}

else {

var nestedObj = obj.getElementsByTagName(OBJECT)[0];

if (nestedObj) {

var c = nestedObj.childNodes;

if (c) {

var cl = c.length;

for (var i = 0; i < cl; i++) {

if (!(c[i].nodeType == 1 && c[i].nodeName == "PARAM") && !(c[i].nodeType == 8)) {

ac.appendChild(c[i].cloneNode(true));

}

}

}

}

}

return ac;

}

/\* Cross-browser dynamic SWF creation

\*/

function createSWF(attObj, parObj, id) {

var r, el = getElementById(id);

if (ua.wk && ua.wk < 312) { return r; }

if (el) {

if (typeof attObj.id == UNDEF) { // if no 'id' is defined for the object element, it will inherit the 'id' from the alternative content

attObj.id = id;

}

if (ua.ie && ua.win) { // Internet Explorer + the HTML object element + W3C DOM methods do not combine: fall back to outerHTML

var att = "";

for (var i in attObj) {

if (attObj[i] != Object.prototype[i]) { // filter out prototype additions from other potential libraries

if (i.toLowerCase() == "data") {

parObj.movie = attObj[i];

}

else if (i.toLowerCase() == "styleclass") { // 'class' is an ECMA4 reserved keyword

att += ' class="' + attObj[i] + '"';

}

else if (i.toLowerCase() != "classid") {

att += ' ' + i + '="' + attObj[i] + '"';

}

}

}

var par = "";

for (var j in parObj) {

if (parObj[j] != Object.prototype[j]) { // filter out prototype additions from other potential libraries

par += '<param name="' + j + '" value="' + parObj[j] + '" />';

}

}

el.outerHTML = '<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"' + att + '>' + par + '</object>';

objIdArr[objIdArr.length] = attObj.id; // stored to fix object 'leaks' on unload (dynamic publishing only)

r = getElementById(attObj.id);

}

else { // well-behaving browsers

var o = createElement(OBJECT);

o.setAttribute("type", FLASH\_MIME\_TYPE);

for (var m in attObj) {

if (attObj[m] != Object.prototype[m]) { // filter out prototype additions from other potential libraries

if (m.toLowerCase() == "styleclass") { // 'class' is an ECMA4 reserved keyword

o.setAttribute("class", attObj[m]);

}

else if (m.toLowerCase() != "classid") { // filter out IE specific attribute

o.setAttribute(m, attObj[m]);

}

}

}

for (var n in parObj) {

if (parObj[n] != Object.prototype[n] && n.toLowerCase() != "movie") { // filter out prototype additions from other potential libraries and IE specific param element

createObjParam(o, n, parObj[n]);

}

}

el.parentNode.replaceChild(o, el);

r = o;

}

}

return r;

}

function createObjParam(el, pName, pValue) {

var p = createElement("param");

p.setAttribute("name", pName);

p.setAttribute("value", pValue);

el.appendChild(p);

}

/\* Cross-browser SWF removal

- Especially needed to safely and completely remove a SWF in Internet Explorer

\*/

function removeSWF(id) {

var obj = getElementById(id);

if (obj && obj.nodeName == "OBJECT") {

if (ua.ie && ua.win) {

obj.style.display = "none";

(function(){

if (obj.readyState == 4) {

removeObjectInIE(id);

}

else {

setTimeout(arguments.callee, 10);

}

})();

}

else {

obj.parentNode.removeChild(obj);

}

}

}

function removeObjectInIE(id) {

var obj = getElementById(id);

if (obj) {

for (var i in obj) {

if (typeof obj[i] == "function") {

obj[i] = null;

}

}

obj.parentNode.removeChild(obj);

}

}

/\* Functions to optimize JavaScript compression

\*/

function getElementById(id) {

var el = null;

try {

el = doc.getElementById(id);

}

catch (e) {}

return el;

}

function createElement(el) {

return doc.createElement(el);

}

/\* Updated attachEvent function for Internet Explorer

- Stores attachEvent information in an Array, so on unload the detachEvent functions can be called to avoid memory leaks

\*/

function addListener(target, eventType, fn) {

target.attachEvent(eventType, fn);

listenersArr[listenersArr.length] = [target, eventType, fn];

}

/\* Flash Player and SWF content version matching

\*/

function hasPlayerVersion(rv) {

var pv = ua.pv, v = rv.split(".");

v[0] = parseInt(v[0], 10);

v[1] = parseInt(v[1], 10) || 0; // supports short notation, e.g. "9" instead of "9.0.0"

v[2] = parseInt(v[2], 10) || 0;

return (pv[0] > v[0] || (pv[0] == v[0] && pv[1] > v[1]) || (pv[0] == v[0] && pv[1] == v[1] && pv[2] >= v[2])) ? true : false;

}

/\* Cross-browser dynamic CSS creation

- Based on Bobby van der Sluis' solution: http://www.bobbyvandersluis.com/articles/dynamicCSS.php

\*/

function createCSS(sel, decl, media, newStyle) {

if (ua.ie && ua.mac) { return; }

var h = doc.getElementsByTagName("head")[0];

if (!h) { return; } // to also support badly authored HTML pages that lack a head element

var m = (media && typeof media == "string") ? media : "screen";

if (newStyle) {

dynamicStylesheet = null;

dynamicStylesheetMedia = null;

}

if (!dynamicStylesheet || dynamicStylesheetMedia != m) {

// create dynamic stylesheet + get a global reference to it

var s = createElement("style");

s.setAttribute("type", "text/css");

s.setAttribute("media", m);

dynamicStylesheet = h.appendChild(s);

if (ua.ie && ua.win && typeof doc.styleSheets != UNDEF && doc.styleSheets.length > 0) {

dynamicStylesheet = doc.styleSheets[doc.styleSheets.length - 1];

}

dynamicStylesheetMedia = m;

}

// add style rule

if (ua.ie && ua.win) {

if (dynamicStylesheet && typeof dynamicStylesheet.addRule == OBJECT) {

dynamicStylesheet.addRule(sel, decl);

}

}

else {

if (dynamicStylesheet && typeof doc.createTextNode != UNDEF) {

dynamicStylesheet.appendChild(doc.createTextNode(sel + " {" + decl + "}"));

}

}

}

function setVisibility(id, isVisible) {

if (!autoHideShow) { return; }

var v = isVisible ? "visible" : "hidden";

if (isDomLoaded && getElementById(id)) {

getElementById(id).style.visibility = v;

}

else {

createCSS("#" + id, "visibility:" + v);

}

}

/\* Filter to avoid XSS attacks

\*/

function urlEncodeIfNecessary(s) {

var regex = /[\\\"<>\.;]/;

var hasBadChars = regex.exec(s) != null;

return hasBadChars && typeof encodeURIComponent != UNDEF ? encodeURIComponent(s) : s;

}

/\* Release memory to avoid memory leaks caused by closures, fix hanging audio/video threads and force open sockets/NetConnections to disconnect (Internet Explorer only)

\*/

var cleanup = function() {

if (ua.ie && ua.win) {

window.attachEvent("onunload", function() {

// remove listeners to avoid memory leaks

var ll = listenersArr.length;

for (var i = 0; i < ll; i++) {

listenersArr[i][0].detachEvent(listenersArr[i][1], listenersArr[i][2]);

}

// cleanup dynamically embedded objects to fix audio/video threads and force open sockets and NetConnections to disconnect

var il = objIdArr.length;

for (var j = 0; j < il; j++) {

removeSWF(objIdArr[j]);

}

// cleanup library's main closures to avoid memory leaks

for (var k in ua) {

ua[k] = null;

}

ua = null;

for (var l in swfobject) {

swfobject[l] = null;

}

swfobject = null;

});

}

}();

return {

/\* Public API

- Reference: http://code.google.com/p/swfobject/wiki/documentation

\*/

registerObject: function(objectIdStr, swfVersionStr, xiSwfUrlStr, callbackFn) {

if (ua.w3 && objectIdStr && swfVersionStr) {

var regObj = {};

regObj.id = objectIdStr;

regObj.swfVersion = swfVersionStr;

regObj.expressInstall = xiSwfUrlStr;

regObj.callbackFn = callbackFn;

regObjArr[regObjArr.length] = regObj;

setVisibility(objectIdStr, false);

}

else if (callbackFn) {

callbackFn({success:false, id:objectIdStr});

}

},

getObjectById: function(objectIdStr) {

if (ua.w3) {

return getObjectById(objectIdStr);

}

},

embedSWF: function(swfUrlStr, replaceElemIdStr, widthStr, heightStr, swfVersionStr, xiSwfUrlStr, flashvarsObj, parObj, attObj, callbackFn) {

var callbackObj = {success:false, id:replaceElemIdStr};

if (ua.w3 && !(ua.wk && ua.wk < 312) && swfUrlStr && replaceElemIdStr && widthStr && heightStr && swfVersionStr) {

setVisibility(replaceElemIdStr, false);

addDomLoadEvent(function() {

widthStr += ""; // auto-convert to string

heightStr += "";

var att = {};

if (attObj && typeof attObj === OBJECT) {

for (var i in attObj) { // copy object to avoid the use of references, because web authors often reuse attObj for multiple SWFs

att[i] = attObj[i];

}

}

att.data = swfUrlStr;

att.width = widthStr;

att.height = heightStr;

var par = {};

if (parObj && typeof parObj === OBJECT) {

for (var j in parObj) { // copy object to avoid the use of references, because web authors often reuse parObj for multiple SWFs

par[j] = parObj[j];

}

}

if (flashvarsObj && typeof flashvarsObj === OBJECT) {

for (var k in flashvarsObj) { // copy object to avoid the use of references, because web authors often reuse flashvarsObj for multiple SWFs

if (typeof par.flashvars != UNDEF) {

par.flashvars += "&" + k + "=" + flashvarsObj[k];

}

else {

par.flashvars = k + "=" + flashvarsObj[k];

}

}

}

if (hasPlayerVersion(swfVersionStr)) { // create SWF

var obj = createSWF(att, par, replaceElemIdStr);

if (att.id == replaceElemIdStr) {

setVisibility(replaceElemIdStr, true);

}

callbackObj.success = true;

callbackObj.ref = obj;

}

else if (xiSwfUrlStr && canExpressInstall()) { // show Adobe Express Install

att.data = xiSwfUrlStr;

showExpressInstall(att, par, replaceElemIdStr, callbackFn);

return;

}

else { // show alternative content

setVisibility(replaceElemIdStr, true);

}

if (callbackFn) { callbackFn(callbackObj); }

});

}

else if (callbackFn) { callbackFn(callbackObj); }

},

switchOffAutoHideShow: function() {

autoHideShow = false;

},

ua: ua,

getFlashPlayerVersion: function() {

return { major:ua.pv[0], minor:ua.pv[1], release:ua.pv[2] };

},

hasFlashPlayerVersion: hasPlayerVersion,

createSWF: function(attObj, parObj, replaceElemIdStr) {

if (ua.w3) {

return createSWF(attObj, parObj, replaceElemIdStr);

}

else {

return undefined;

}

},

showExpressInstall: function(att, par, replaceElemIdStr, callbackFn) {

if (ua.w3 && canExpressInstall()) {

showExpressInstall(att, par, replaceElemIdStr, callbackFn);

}

},

removeSWF: function(objElemIdStr) {

if (ua.w3) {

removeSWF(objElemIdStr);

}

},

createCSS: function(selStr, declStr, mediaStr, newStyleBoolean) {

if (ua.w3) {

createCSS(selStr, declStr, mediaStr, newStyleBoolean);

}

},

addDomLoadEvent: addDomLoadEvent,

addLoadEvent: addLoadEvent,

getQueryParamValue: function(param) {

var q = doc.location.search || doc.location.hash;

if (q) {

if (/\?/.test(q)) { q = q.split("?")[1]; } // strip question mark

if (param == null) {

return urlEncodeIfNecessary(q);

}

var pairs = q.split("&");

for (var i = 0; i < pairs.length; i++) {

if (pairs[i].substring(0, pairs[i].indexOf("=")) == param) {

return urlEncodeIfNecessary(pairs[i].substring((pairs[i].indexOf("=") + 1)));

}

}

}

return "";

},

// For internal usage only

expressInstallCallback: function() {

if (isExpressInstallActive) {

var obj = getElementById(EXPRESS\_INSTALL\_ID);

if (obj && storedAltContent) {

obj.parentNode.replaceChild(storedAltContent, obj);

if (storedAltContentId) {

setVisibility(storedAltContentId, true);

if (ua.ie && ua.win) { storedAltContent.style.display = "block"; }

}

if (storedCallbackFn) { storedCallbackFn(storedCallbackObj); }

}

isExpressInstallActive = false;

}

}

};

}();

<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<actionScriptProperties analytics="false" mainApplicationPath="FlashPlayerDemo.mxml" projectUUID="5cee88e5-794a-4e88-9090-b09fb639577b" version="10">

<compiler additionalCompilerArguments="-locale en\_US" autoRSLOrdering="true" copyDependentFiles="true" fteInMXComponents="false" generateAccessible="true" htmlExpressInstall="true" htmlGenerate="true" htmlHistoryManagement="true" htmlPlayerVersionCheck="true" includeNetmonSwc="false" outputFolderPath="bin-debug" removeUnusedRSL="true" sourceFolderPath="src" strict="true" targetPlayerVersion="0.0.0" useApolloConfig="false" useDebugRSLSwfs="true" verifyDigests="true" warn="true">

<compilerSourcePath/>

<libraryPath defaultLinkType="0">

<libraryPathEntry kind="4" path="">

<excludedEntries>

<libraryPathEntry kind="3" linkType="1" path="${PROJECT\_FRAMEWORKS}/libs/flex.swc" useDefaultLinkType="false"/>

<libraryPathEntry kind="3" linkType="1" path="${PROJECT\_FRAMEWORKS}/libs/core.swc" useDefaultLinkType="false"/>

</excludedEntries>

</libraryPathEntry>

<libraryPathEntry kind="1" linkType="1" path="libs"/>

</libraryPath>

<sourceAttachmentPath/>

</compiler>

<applications>

<application path="FlashPlayerDemo.mxml"/>

</applications>

<modules/>

<buildCSSFiles/>

<flashCatalyst validateFlashCatalystCompatibility="false"/>

</actionScriptProperties>

<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<flexProperties enableServiceManager="false" flexServerFeatures="0" flexServerType="0" toolCompile="true" useServerFlexSDK="false" version="2"/>

<?xml version="1.0" encoding="UTF-8"?>

<projectDescription>

<name>FlashPlayerDemo</name>

<comment></comment>

<projects>

</projects>

<buildSpec>

<buildCommand>

<name>com.adobe.flexbuilder.project.flexbuilder</name>

<arguments>

</arguments>

</buildCommand>

</buildSpec>

<natures>

<nature>com.adobe.flexbuilder.project.flexnature</nature>

<nature>com.adobe.flexbuilder.project.actionscriptnature</nature>

</natures>

</projectDescription>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<!-- saved from url=(0014)about:internet -->

<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">

<!--

Smart developers always View Source.

This application was built using Adobe Flex, an open source framework

for building rich Internet applications that get delivered via the

Flash Player or to desktops via Adobe AIR.

Learn more about Flex at http://flex.org

// -->

<head>

<title>${title}</title>

<meta name="google" value="notranslate" />

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<!-- Include CSS to eliminate any default margins/padding and set the height of the html element and

the body element to 100%, because Firefox, or any Gecko based browser, interprets percentage as

the percentage of the height of its parent container, which has to be set explicitly. Fix for

Firefox 3.6 focus border issues. Initially, don't display flashContent div so it won't show

if JavaScript disabled.

-->

<style type="text/css" media="screen">

html, body { height:100%; }

body { margin:0; padding:0; overflow:auto; text-align:center;

background-color: ${bgcolor}; }

object:focus { outline:none; }

#flashContent { display:none; }

</style>

<!-- Enable Browser History by replacing useBrowserHistory tokens with two hyphens -->

<!-- BEGIN Browser History required section ${useBrowserHistory}>

<link rel="stylesheet" type="text/css" href="history/history.css" />

<script type="text/javascript" src="history/history.js"></script>

<!${useBrowserHistory} END Browser History required section -->

<script type="text/javascript" src="swfobject.js"></script>

<script type="text/javascript">

// For version detection, set to min. required Flash Player version, or 0 (or 0.0.0), for no version detection.

var swfVersionStr = "${version\_major}.${version\_minor}.${version\_revision}";

// To use express install, set to playerProductInstall.swf, otherwise the empty string.

var xiSwfUrlStr = "${expressInstallSwf}";

var flashvars = {};

var params = {};

params.quality = "high";

params.bgcolor = "${bgcolor}";

params.allowscriptaccess = "sameDomain";

params.allowfullscreen = "true";

var attributes = {};

attributes.id = "${application}";

attributes.name = "${application}";

attributes.align = "middle";

swfobject.embedSWF(

"${swf}.swf", "flashContent",

"${width}", "${height}",

swfVersionStr, xiSwfUrlStr,

flashvars, params, attributes);

// JavaScript enabled so display the flashContent div in case it is not replaced with a swf object.

swfobject.createCSS("#flashContent", "display:block;text-align:left;");

</script>

</head>

<body>

<!-- SWFObject's dynamic embed method replaces this alternative HTML content with Flash content when enough

JavaScript and Flash plug-in support is available. The div is initially hidden so that it doesn't show

when JavaScript is disabled.

-->

<div id="flashContent">

<p>

To view this page ensure that Adobe Flash Player version

${version\_major}.${version\_minor}.${version\_revision} or greater is installed.

</p>

<script type="text/javascript">

var pageHost = ((document.location.protocol == "https:") ? "https://" : "http://");

document.write("<a href='http://www.adobe.com/go/getflashplayer'><img src='"

+ pageHost + "www.adobe.com/images/shared/download\_buttons/get\_flash\_player.gif' alt='Get Adobe Flash player' /></a>" );

</script>

</div>

<noscript>

<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" width="${width}" height="${height}" id="${application}">

<param name="movie" value="${swf}.swf" />

<param name="quality" value="high" />

<param name="bgcolor" value="${bgcolor}" />

<param name="allowScriptAccess" value="sameDomain" />

<param name="allowFullScreen" value="true" />

<!--[if !IE]>-->

<object type="application/x-shockwave-flash" data="${swf}.swf" width="${width}" height="${height}">

<param name="quality" value="high" />

<param name="bgcolor" value="${bgcolor}" />

<param name="allowScriptAccess" value="sameDomain" />

<param name="allowFullScreen" value="true" />

<!--<![endif]-->

<!--[if gte IE 6]>-->

<p>

Either scripts and active content are not permitted to run or Adobe Flash Player version

${version\_major}.${version\_minor}.${version\_revision} or greater is not installed.

</p>

<!--<![endif]-->

<a href="http://www.adobe.com/go/getflashplayer">

<img src="http://www.adobe.com/images/shared/download\_buttons/get\_flash\_player.gif" alt="Get Adobe Flash Player" />

</a>

<!--[if !IE]>-->

</object>

<!--<![endif]-->

</object>

</noscript>

</body>

</html>

<html>

<head>

<META HTTP-EQUIV="Pragma" CONTENT="no-cache">

<META HTTP-EQUIV="Expires" CONTENT="-1">

</head>

<body>

<script>

function processUrl()

{

var pos = url.indexOf("?");

url = pos != -1 ? url.substr(pos + 1) : "";

if (!parent.\_ie\_firstload) {

parent.BrowserHistory.setBrowserURL(url);

try {

parent.BrowserHistory.browserURLChange(url);

} catch(e) { }

} else {

parent.\_ie\_firstload = false;

}

}

var url = document.location.href;

processUrl();

document.write(encodeURIComponent(url));

</script>

Hidden frame for Browser History support.

</body>

</html>

<?xml version="1.0" encoding="utf-8" standalone="yes"?>

<content Name="TEAM MEMBERS">

<gallery Name="MANAGEMENT">

<item Image="team/gallery1/image1.jpg">

<copy><![CDATA[<font color="#FFFFFF" size="20">Joe Blogs</font><br><br><font color="#FFFFFF" size="11">Position: Chief Executive Officer</font><br>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.<br><br><font color="#FFFFFF">Contact Details:</font><br>T: +27 012 123 1234<br>F: +27 012 123 1234<br>E: <a href="mailto:name@company.com">name@company.com</a>]]></copy>

</item>

<item Image="team/gallery1/image2.jpg">

<copy><![CDATA[<font color="#FFFFFF" size="20">Tom Smith</font><br><br><font color="#FFFFFF" size="11">Position: Director</font><br>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.<br><br><font color="#FFFFFF">Contact Details:</font><br>T: +27 012 123 1234<br>F: +27 012 123 1234<br>E: <a href="mailto:name@company.com">name@company.com</a>]]></copy>

</item>

<item Image="team/gallery1/image3.jpg">

<copy><![CDATA[<font color="#FFFFFF" size="20">Jane Doe</font><br><br><font color="#FFFFFF" size="11">Position: Creative Manager</font><br>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.<br><br><font color="#FFFFFF">Contact Details:</font><br>T: +27 012 123 1234<br>F: +27 012 123 1234<br>E: <a href="mailto:name@company.com">name@company.com</a>]]></copy>

</item>

<item Image="team/gallery1/image4.jpg">

<copy><![CDATA[<font color="#FFFFFF" size="20">John Smith</font><br><br><font color="#FFFFFF" size="11">Position: Production Manager</font><br>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.<br><br><font color="#FFFFFF">Contact Details:</font><br>T: +27 012 123 1234<br>F: +27 012 123 1234<br>E: <a href="mailto:name@company.com">name@company.com</a>]]></copy>

</item>

<item Image="team/gallery1/image1.jpg">

<copy><![CDATA[<font color="#FFFFFF" size="20">Mary Jane</font><br><br><font color="#FFFFFF" size="11">Position: Technical Administrator</font><br>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.<br><br><font color="#FFFFFF">Contact Details:</font><br>T: +27 012 123 1234<br>F: +27 012 123 1234<br>E: <a href="mailto:name@company.com">name@company.com</a>]]></copy>

</item>

<item Image="team/gallery1/image2.jpg">

<copy><![CDATA[<font color="#FFFFFF" size="20">James Smith</font><br><br><font color="#FFFFFF" size="11">Position: Human Resources</font><br>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.<br><br><font color="#FFFFFF">Contact Details:</font><br>T: +27 012 123 1234<br>F: +27 012 123 1234<br>E: <a href="mailto:name@company.com">name@company.com</a>]]></copy>

</item>

<item Image="team/gallery1/image3.jpg">

<copy><![CDATA[<font color="#FFFFFF" size="20">John Doe</font><br><br><font color="#FFFFFF" size="11">Position: Accounts Manager</font><br>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.<br><br><font color="#FFFFFF">Contact Details:</font><br>T: +27 012 123 1234<br>F: +27 012 123 1234<br>E: <a href="mailto:name@company.com">name@company.com</a>]]></copy>

</item>

</gallery>

<gallery Name="DEVELOPERS">

<item Image="team/gallery2/image1.jpg">

<copy><![CDATA[<font color="#FFFFFF" size="20">James Smith</font><br><br><font color="#FFFFFF" size="11">Position: Lead Developer</font><br>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.<br><br><font color="#FFFFFF">Contact Details:</font><br>T: +27 012 123 1234<br>F: +27 012 123 1234<br>E: <a href="mailto:name@company.com">name@company.com</a>]]></copy>

</item>

<item Image="team/gallery2/image2.jpg">

<copy><![CDATA[<font color="#FFFFFF" size="20">John Doe</font><br><br><font color="#FFFFFF" size="11">Position: System Administrator</font><br>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.<br><br><font color="#FFFFFF">Contact Details:</font><br>T: +27 012 123 1234<br>F: +27 012 123 1234<br>E: <a href="mailto:name@company.com">name@company.com</a>]]></copy>

</item>

<item Image="team/gallery2/image3.jpg">

<copy><![CDATA[<font color="#FFFFFF" size="20">Tom Smith</font><br><br><font color="#FFFFFF" size="11">Position: Senior Developer</font><br>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.<br><br><font color="#FFFFFF">Contact Details:</font><br>T: +27 012 123 1234<br>F: +27 012 123 1234<br>E: <a href="mailto:name@company.com">name@company.com</a>]]></copy>

</item>

<item Image="team/gallery2/image4.jpg">

<copy><![CDATA[<font color="#FFFFFF" size="20">John Smith</font><br><br><font color="#FFFFFF" size="11">Position: Senior Developer</font><br>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.<br><br><font color="#FFFFFF">Contact Details:</font><br>T: +27 012 123 1234<br>F: +27 012 123 1234<br>E: <a href="mailto:name@company.com">name@company.com</a>]]></copy>

</item>

<item Image="team/gallery2/image1.jpg">

<copy><![CDATA[<font color="#FFFFFF" size="20">Mary Jane</font><br><br><font color="#FFFFFF" size="11">Position: Developer</font><br>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.<br><br><font color="#FFFFFF">Contact Details:</font><br>T: +27 012 123 1234<br>F: +27 012 123 1234<br>E: <a href="mailto:name@company.com">name@company.com</a>]]></copy>

</item>

</gallery>

<gallery Name="DESIGNERS">

<item Image="team/gallery3/image1.jpg">

<copy><![CDATA[<font color="#FFFFFF" size="20">Jane Doe</font><br><br><font color="#FFFFFF" size="11">Position: Lead Designer</font><br>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.<br><br><font color="#FFFFFF">Contact Details:</font><br>T: +27 012 123 1234<br>F: +27 012 123 1234<br>E: <a href="mailto:name@company.com">name@company.com</a>]]></copy>

</item>

<item Image="team/gallery3/image2.jpg">

<copy><![CDATA[<font color="#FFFFFF" size="20">John Smith</font><br><br><font color="#FFFFFF" size="11">Position: Senior Designer</font><br>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.<br><br><font color="#FFFFFF">Contact Details:</font><br>T: +27 012 123 1234<br>F: +27 012 123 1234<br>E: <a href="mailto:name@company.com">name@company.com</a>]]></copy>

</item>

<item Image="team/gallery3/image3.jpg">

<copy><![CDATA[<font color="#FFFFFF" size="20">Mary Jane</font><br><br><font color="#FFFFFF" size="11">Position: Designer</font><br>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.<br><br><font color="#FFFFFF">Contact Details:</font><br>T: +27 012 123 1234<br>F: +27 012 123 1234<br>E: <a href="mailto:name@company.com">name@company.com</a>]]></copy>

</item>

</gallery>

</content>

{

ExternalInterface.addCallback("sayCallBack",callBack);

}

public function callBack(str:String):String{

return "Hello "+str;

}

protected function button1\_clickHandler(event:MouseEvent):void

{

ExternalInterface.call('sayHelloWorld',' Jim');

var array:Array = new Array();

array.push("a", "b","c","d","e");

myIFrame.callIFrameFunction('jsMethod',array,function(str:String):void{

label.text=str;

});

}

public static function str():String{

return "aaa";

}

]]>

</fx:Script>

<s:VGroup width="100%" height="100%">

<s:BorderContainer width="100%" height="300" backgroundColor="#00ffaa">

<s:HGroup width="100%">

<s:Button label="调用js" click="button1\_clickHandler(event)"/>

<mx:Button x="169.5" y="162" label="点击" fontSize="14" id="button"/>

<mx:TextInput id="asInput" x="122" y="76"/>

<s:Label id="label"/>

</s:HGroup>

</s:BorderContainer>

<s:BorderContainer width="100%" height="300">

<code:IFrame id="myIFrame" source="com/MyHtml.html" width="100%" height="100%"/>

</s:BorderContainer>

</s:VGroup>

</s:Application>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" lang="zh-CN" xml:lang="zh-CN">

<head>

<title>test-3</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<style type="text/css" media="screen">

html, body { height:100%; background-color: #ffffff;}

body { margin:0; padding:0; overflow:hidden; }

#flashContent { width:100%; height:100%; }

</style>

<script type="text/javascript">

function AsCallJs(str){

trace(str);

}

</script>

</head>

<body>

<div id="flashContent">

<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" width="550" height="400" id="test-3" align="middle">

<param name="movie" value="test-3.swf" />

<param name="quality" value="high" />

<param name="bgcolor" value="#ffffff" />

<param name="play" value="true" />

<param name="loop" value="true" />

<param name="wmode" value="window" />

<param name="scale" value="showall" />

<param name="menu" value="true" />

<param name="devicefont" value="false" />

<param name="salign" value="" />

<param name="allowScriptAccess" value="sameDomain" />

<!--[if !IE]>-->

<object type="application/x-shockwave-flash" data="test-3.swf" width="550" height="400">

<param name="movie" value="test-3.swf" />

<param name="quality" value="high" />

<param name="bgcolor" value="#ffffff" />

<param name="play" value="true" />

<param name="loop" value="true" />

<param name="wmode" value="window" />

<param name="scale" value="showall" />

<param name="menu" value="true" />

<param name="devicefont" value="false" />

<param name="salign" value="" />

<param name="allowScriptAccess" value="sameDomain" />

<!--<![endif]-->

<a href="http://www.adobe.com/go/getflash">

<img src="http://www.adobe.com/images/shared/download\_buttons/get\_flash\_player.gif" alt="获得 Adobe Flash Player" />

</a>

<!--[if !IE]>-->

</object>

<!--<![endif]-->

</object>

<button type="button" name="button">123</button>

</div>

</body>

</html>

<?xml version="1.0" encoding="utf-8" standalone="yes"?>

<content Delay="10000">

<bg Image="backgrounds/image1.jpg"/>

<bg Image="backgrounds/image2.jpg"/>

<bg Image="backgrounds/image3.jpg"/>

<bg Image="backgrounds/image4.jpg"/>

<bg Image="backgrounds/image5.jpg"/>

<bg Image="backgrounds/image6.jpg"/>

</content>

<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<actionScriptProperties analytics="false" mainApplicationPath="tt.as" projectUUID="f8f4baf0-4be4-4120-8f54-891027e9d102" version="11">

<compiler additionalCompilerArguments="-locale en\_US" advancedTelemetry="false" autoRSLOrdering="true" copyDependentFiles="true" fteInMXComponents="false" generateAccessible="false" htmlExpressInstall="true" htmlGenerate="true" htmlHistoryManagement="true" htmlPlayerVersionCheck="true" includeNetmonSwc="false" outputFolderPath="bin-debug" removeUnusedRSL="true" sourceFolderPath="src" strict="true" targetPlayerVersion="0.0.0" useApolloConfig="false" useDebugRSLSwfs="true" useFlashSDK="true" verifyDigests="true" warn="true">

<compilerSourcePath/>

<libraryPath defaultLinkType="0">

<libraryPathEntry kind="4" path="">

<excludedEntries>

<libraryPathEntry kind="3" linkType="1" path="${PROJECT\_FRAMEWORKS}/libs/automation\_charts.swc" useDefaultLinkType="false"/>

<libraryPathEntry kind="1" linkType="1" path="${PROJECT\_FRAMEWORKS}/locale/{locale}"/>

<libraryPathEntry kind="3" linkType="1" path="${PROJECT\_FRAMEWORKS}/libs/advancedgrids.swc" useDefaultLinkType="false"/>

<libraryPathEntry kind="3" linkType="1" path="${PROJECT\_FRAMEWORKS}/libs/qtp.swc" useDefaultLinkType="false"/>

<libraryPathEntry kind="3" linkType="1" path="${PROJECT\_FRAMEWORKS}/libs/automation\_air.swc" useDefaultLinkType="false"/>

<libraryPathEntry kind="3" linkType="1" path="${PROJECT\_FRAMEWORKS}/libs/charts.swc" useDefaultLinkType="false"/>

<libraryPathEntry kind="3" linkType="1" path="${PROJECT\_FRAMEWORKS}/libs/framework.swc" useDefaultLinkType="false"/>

<libraryPathEntry kind="3" linkType="1" path="${PROJECT\_FRAMEWORKS}/libs/mx/mx.swc" useDefaultLinkType="false"/>

<libraryPathEntry kind="3" linkType="1" path="${PROJECT\_FRAMEWORKS}/libs/netmon.swc" useDefaultLinkType="false"/>

<libraryPathEntry kind="3" linkType="1" path="${PROJECT\_FRAMEWORKS}/libs/spark.swc" useDefaultLinkType="false"/>

<libraryPathEntry kind="3" linkType="1" path="${PROJECT\_FRAMEWORKS}/libs/sparkskins.swc" useDefaultLinkType="false"/>

<libraryPathEntry kind="3" linkType="1" path="${PROJECT\_FRAMEWORKS}/libs/rpc.swc" useDefaultLinkType="false"/>

<libraryPathEntry kind="3" linkType="1" path="${PROJECT\_FRAMEWORKS}/libs/videoPlayer.swc" useDefaultLinkType="false"/>

<libraryPathEntry kind="3" linkType="1" path="${PROJECT\_FRAMEWORKS}/libs/qtp\_air.swc" useDefaultLinkType="false"/>

<libraryPathEntry kind="3" linkType="1" path="${PROJECT\_FRAMEWORKS}/libs/datavisualization.swc" useDefaultLinkType="false"/>

<libraryPathEntry kind="3" linkType="1" path="${PROJECT\_FRAMEWORKS}/libs/spark\_dmv.swc" useDefaultLinkType="false"/>

<libraryPathEntry kind="3" linkType="1" path="${PROJECT\_FRAMEWORKS}/libs/automation.swc" useDefaultLinkType="false"/>

<libraryPathEntry kind="3" linkType="1" path="${PROJECT\_FRAMEWORKS}/libs/flash-integration.swc" useDefaultLinkType="false"/>

<libraryPathEntry kind="3" linkType="1" path="${PROJECT\_FRAMEWORKS}/libs/automation\_dmv.swc" useDefaultLinkType="false"/>

<libraryPathEntry kind="3" linkType="1" path="${PROJECT\_FRAMEWORKS}/libs/automation\_flashflexkit.swc" useDefaultLinkType="false"/>

<libraryPathEntry kind="3" linkType="1" path="${PROJECT\_FRAMEWORKS}/libs/automation\_agent.swc" useDefaultLinkType="false"/>

</excludedEntries>

</libraryPathEntry>

</libraryPath>

<sourceAttachmentPath/>

</compiler>

<applications/>

<modules/>

<workers/>

<buildCSSFiles/>

<flashCatalyst validateFlashCatalystCompatibility="false"/>

</actionScriptProperties>

package controls {

import flash.events.TimerEvent;

import flash.utils.Timer;

import mx.core.UIComponent;

import mx.events.FlexEvent;

import mx.styles.CSSStyleDeclaration;

import mx.styles.StyleManager;

[Style(name="tickColor",type="uint",format="Color",inherit="no")]

/\*\*

\* Creates a spinning "loader" component that is sort of an indeterminate progress bar.

\* @author jhawkes

\*

\*/

class Spinner extends UIComponent {

private static var STYLE\_TICK\_COLOR:String = "tickColor";

private var tickColorChanged:Boolean;

private static var classConstructed:Boolean = classConstruct();

// Make sure we create the ticks the first time updateDisplayList is called

private var creation:Boolean = true;

private var fadeTimer:Timer;

private var \_isPlaying:Boolean;

private var \_numTicks:int = 12;

private var numTicksChanged:Boolean;

private var \_size:Number = 30;

private var sizeChanged:Boolean;

private var \_tickWidth:Number = 3;

private var tickWidthChanged:Boolean;

private var \_speed:int = 1000;

[Bindable] public var fadeSpeed:int = 600;

public var autoPlay:Boolean = true;

public function Spinner() {

super();

addEventListener(FlexEvent.CREATION\_COMPLETE, handleCreationComplete);

}

private function handleCreationComplete(e:FlexEvent):void {

removeEventListener(FlexEvent.CREATION\_COMPLETE, handleCreationComplete);

if (autoPlay) {

play();

}

}

/\*\*

\* Set the height and width based on the size of the spinner. This should be more robust, but oh well.

\*/

override protected function measure():void {

super.measure();

width = \_size;

height = \_size;

}

/\*\*

\* Override the updateDisplayList method

\*/

override protected function updateDisplayList(unscaledWidth:Number, unscaledHeight:Number):void {

if (tickColorChanged || numTicksChanged || sizeChanged || tickWidthChanged || creation) {

creation = false;

// Find out whether it's playing so we can restart it later if we need to

var wasPlaying:Boolean = \_isPlaying;

// stop the spinning

stop();

// Remove all children

for (var i:int = numChildren - 1; i >= 0; i--) {

removeChildAt(i);

}

// Re-create the children

var radius:Number = size / 2;

var angle:Number = 2 \* Math.PI / \_numTicks; // The angle between each tick

var tickWidth:Number = (\_tickWidth != -1) ? \_tickWidth : size / 10;

var tickColor:uint = getStyle(STYLE\_TICK\_COLOR);

var currentAngle:Number = 0;

for (var j:int = 0; j < \_numTicks; j++) {

var xStart:Number = radius + Math.sin(currentAngle) \* ((\_numTicks + 2) \* tickWidth / 2 / Math.PI);

var yStart:Number = radius - Math.cos(currentAngle) \* ((\_numTicks + 2) \* tickWidth / 2 / Math.PI);

var xEnd:Number = radius + Math.sin(currentAngle) \* (radius - tickWidth);

var yEnd:Number = radius - Math.cos(currentAngle) \* (radius - tickWidth);

var t:Tick = new Tick(xStart, yStart, xEnd, yEnd, tickWidth, tickColor);

t.alpha = 0.1;

this.addChild(t);

currentAngle += angle;

}

// Start the spinning again if it was playing when this function was called.

if (wasPlaying) {

play();

}

tickColorChanged = false;

numTicksChanged = false;

sizeChanged = false;

tickWidthChanged = false;

}

}

private static function classConstruct():Boolean {

if (!StyleManager.getStyleDeclaration("Spinner")) {

// If there is no CSS definition for StyledRectangle,

// then create one and set the default value.

var newStyleDeclaration:CSSStyleDeclaration = new CSSStyleDeclaration();

newStyleDeclaration.setStyle(STYLE\_TICK\_COLOR, 0x000000);

StyleManager.setStyleDeclaration("Spinner", newStyleDeclaration, true);

}

return true;

}

override public function styleChanged(styleProp:String):void {

if (styleProp == STYLE\_TICK\_COLOR) {

tickColorChanged = true;

invalidateDisplayList();

}

}

/\*\*

\* Begin the circular fading of the ticks.

\*/

public function play():void {

if (! \_isPlaying) {

fadeTimer = new Timer(speed / \_numTicks, 0);

// Anonymous functions are especially useful as simple event handlers

fadeTimer.addEventListener(TimerEvent.TIMER, function (e:TimerEvent):void {

var tickNum:int = int(fadeTimer.currentCount % \_numTicks);

if (numChildren > tickNum) {

var tick:Tick = getChildAt(tickNum) as Tick;

tick.fade(fadeSpeed != 1 ? fadeSpeed : speed \* 6 / 10);

}

});

fadeTimer.start();

\_isPlaying = true;

}

}

/\*\*

\* Stop the spinning.

\*/

public function stop():void {

if (fadeTimer != null && fadeTimer.running) {

\_isPlaying = false;

fadeTimer.stop();

}

}

/\*\*

\* The overall diameter of the spinner; also the height and width.

\*/

[Bindable]

public function set size(value:Number):void {

if (value != \_size) {

\_size = value;

sizeChanged = true;

invalidateDisplayList();

invalidateSize();

}

}

public function get size():Number {

return \_size;

}

/\*\*

\* The number of 'spokes' on the spinner.

\*/

[Bindable]

public function set numTicks(value:int):void {

if (value != \_numTicks) {

\_numTicks = value;

numTicksChanged = true;

invalidateDisplayList();

}

}

public function get numTicks():int {

return \_numTicks;

}

/\*\*

\* The width of the 'spokes' on the spinner.

\*/

[Bindable]

public function set tickWidth(value:int):void {

if (value != \_tickWidth) {

\_tickWidth = value;

tickWidthChanged = true;

invalidateDisplayList();

}

}

public function get tickWidth():int {

return \_tickWidth;

}

/\*\*

\* The duration (in milliseconds) that it takes for the spinner to make one revolution.

\*/

[Bindable]

public function set speed(value:int):void {

if (value != \_speed) {

\_speed = value;

if (fadeTimer != null) {

fadeTimer.stop();

fadeTimer.delay = value / \_numTicks;

fadeTimer.start();

}

}

}

public function get speed():int {

return \_speed;

}

public function get isPlaying():Boolean {

return \_isPlaying;

}

}

}

<?xml version="1.0" encoding="utf-8" standalone="yes"?>

<content>

<settings>

<logo X="5" Y="6">content/logo.png</logo>

<footer><![CDATA[Supernova XML Website by <font color="#00FFFF"><a href="http://bbs.dayalive.com" target="\_blank">dayalive.com</a></font>]]></footer>

<menu X="160"/>

</settings>

<nav>

<main Name="HOME" Link="home.swf"/>

<main Name="SLIDE SHOW" Link="slideshow.swf"/>

<main Name="TEAM MEMBERS" Link="team.swf"/>

<main Name="MEDIA GALLERY" Link="media\_gallery.swf"/>

<main Name="STANDALONE GALLERIES" Link="content.swf" toLoad="content/content\_about\_standalone.xml">

<sub Name="IMAGE GALLERY" Link="media\_gallery.swf" toLoad="media\_gallery/content\_image.xml"/>

<sub Name="VIDEO GALLERY" Link="media\_gallery.swf" toLoad="media\_gallery/content\_video.xml"/>

<sub Name="AUDIO GALLERY" Link="media\_gallery.swf" toLoad="media\_gallery/content\_audio.xml"/>

<sub Name="FLASH GALLERY" Link="media\_gallery.swf" toLoad="media\_gallery/content\_flash.xml"/>

<sub Name="MIXED MEDIA GALLERY" Link="media\_gallery.swf" toLoad="media\_gallery/content\_mixed.xml"/>

</main>

<main Name="CONTENT WINDOW" Link="content.swf" toLoad="content/content.xml"/>

<main Name="NEWS" Link="news.swf"/>

<main Name="CONTACT US" Link="contact.swf"/>

<main Name="EXTERNAL LINK" Link="http://bbs.dayalive.com"/>

</nav>

</content>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<!-- saved from url=(0014)about:internet -->

<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">

<!--

Smart developers always View Source.

This application was built using Adobe Flex, an open source framework

for building rich Internet applications that get delivered via the

Flash Player or to desktops via Adobe AIR.

Learn more about Flex at http://flex.org

// -->

<head>

<title></title>

<meta name="google" value="notranslate" />

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<!-- Include CSS to eliminate any default margins/padding and set the height of the html element and

the body element to 100%, because Firefox, or any Gecko based browser, interprets percentage as

the percentage of the height of its parent container, which has to be set explicitly. Fix for

Firefox 3.6 focus border issues. Initially, don't display flashContent div so it won't show

if JavaScript disabled.

-->

<style type="text/css" media="screen">

html, body { height:100%; }

body { margin:0; padding:0; overflow:auto; text-align:center;

background-color: #ffffff; }

object:focus { outline:none; }

#flashContent { display:none; }

</style>

<!-- Enable Browser History by replacing useBrowserHistory tokens with two hyphens -->

<!-- BEGIN Browser History required section -->

<link rel="stylesheet" type="text/css" href="history/history.css" />

<script type="text/javascript" src="history/history.js"></script>

<!-- END Browser History required section -->

<script type="text/javascript" src="swfobject.js"></script>

<script type="text/javascript">

// For version detection, set to min. required Flash Player version, or 0 (or 0.0.0), for no version detection.

var swfVersionStr = "11.1.0";

// To use express install, set to playerProductInstall.swf, otherwise the empty string.

var xiSwfUrlStr = "playerProductInstall.swf";

var flashvars = {};

var params = {};

params.quality = "high";

params.bgcolor = "#ffffff";

params.allowscriptaccess = "sameDomain";

params.allowfullscreen = "true";

var attributes = {};

attributes.id = "FlashPlayerDemo";

attributes.name = "FlashPlayerDemo";

attributes.align = "middle";

swfobject.embedSWF(

"FlashPlayerDemo.swf", "flashContent",

"100%", "100%",

swfVersionStr, xiSwfUrlStr,

flashvars, params, attributes);

// JavaScript enabled so display the flashContent div in case it is not replaced with a swf object.

swfobject.createCSS("#flashContent", "display:block;text-align:left;");

</script>

</head>

<body>

<!-- SWFObject's dynamic embed method replaces this alternative HTML content with Flash content when enough

JavaScript and Flash plug-in support is available. The div is initially hidden so that it doesn't show

when JavaScript is disabled.

-->

<div id="flashContent">

<p>

To view this page ensure that Adobe Flash Player version

11.1.0 or greater is installed.

</p>

<script type="text/javascript">

var pageHost = ((document.location.protocol == "https:") ? "https://" : "http://");

document.write("<a href='http://www.adobe.com/go/getflashplayer'><img src='"

+ pageHost + "www.adobe.com/images/shared/download\_buttons/get\_flash\_player.gif' alt='Get Adobe Flash player' /></a>" );

</script>

</div>

<noscript>

<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" width="100%" height="100%" id="FlashPlayerDemo">

<param name="movie" value="FlashPlayerDemo.swf" />

<param name="quality" value="high" />

<param name="bgcolor" value="#ffffff" />

<param name="allowScriptAccess" value="sameDomain" />

<param name="allowFullScreen" value="true" />

<!--[if !IE]>-->

<object type="application/x-shockwave-flash" data="FlashPlayerDemo.swf" width="100%" height="100%">

<param name="quality" value="high" />

<param name="bgcolor" value="#ffffff" />

<param name="allowScriptAccess" value="sameDomain" />

<param name="allowFullScreen" value="true" />

<!--<![endif]-->

<!--[if gte IE 6]>-->

<p>

Either scripts and active content are not permitted to run or Adobe Flash Player version

11.1.0 or greater is not installed.

</p>

<!--<![endif]-->

<a href="http://www.adobe.com/go/getflashplayer">

<img src="http://www.adobe.com/images/shared/download\_buttons/get\_flash\_player.gif" alt="Get Adobe Flash Player" />

</a>

<!--[if !IE]>-->

</object>

<!--<![endif]-->

</object>

</noscript>

</body>

</html>