```
// copyright stevewhetstone.com All rights resereved.
import fl.video.VideoEvent;
import caurina.transitions.Tweener;

var i:uint;// reusable loop variable

// ------------- Initialize functions ----------------------
var playListPath:String="index_video.xml";
var bookMark:String="hello";
var autoPlaysLeft = 15;
this.loaderInfo.addEventListener(Event.COMPLETE, loaderComplete);// used to check for bookmarks and
function loaderComplete(myEvent:Event) {
    if (ExternalInterface.available) {
        var flashVars=this.loaderInfo.parameters;
        if (flashVars.playlist) {
            playListPath="custom/"+flashVars.playlist+".xml";
            reportUsage("someone used the "+flashVars.playlist+" playlist");
        }
        if (flashVars.bookmark) {
            bookMark=flashVars.bookmark;
        }
    }
    loader.load(new URLRequest(playListPath));
}

//standard XML loading process
var xmlData:XML;
var _items:XMLList;
var loader:URLLoader = new URLLoader();
loader.addEventListener(IOErrorEvent.IO_ERROR, ioErrorHandler);//in case it can't find the XML file
loader.addEventListener(Event.COMPLETE, onXMLLoad);

function ioErrorHandler(event:IOErrorEvent):void {
    trace("There was an error loading the XML file: " + event.text);
    // load the default xml file
    loader.load(new URLRequest("index.xml"));
}
function onXMLLoad(event:Event):void {
    XML.ignoreWhitespace=true;
    xmlData=new XML(event.target.data);//extracts the XML data from the loader object
    _items=xmlData.entry;//aim for the first child node
    initializeSite();
}
function initializeSite() {
    // runs on first load only.
    // getbookmark and load xml playlist
    identityInitialize();
    tocInitialize();
    //tocVideoMenuBackground.addEventListener(VideoEvent.COMPLETE, tocVideoEnd);// Detect end of mo
    //tocVideoMenuBackground.volume=.5;
    myPlayer.addEventListener(VideoEvent.PLAYING_STATE_ENTERED, playingState);// when we start play
    myPlayer.addEventListener(VideoEvent.COMPLETE, movieEnd);// Detect end of movie
    myPlayer.playheadUpdateInterval=41;// this makes the seek bar smooth.
    myPlayer.seekBarInterval=41;// this makes the seek bar smooth when scrubbing.
```

```
    myPlayer.seekBarScrubTolerance=1;// this does almost nothing because movies skip to the nearest
    myPlayer.volume=.5;// set it to medium volume
    stage.addEventListener(MouseEvent.MOUSE_MOVE,captureMove);//capture mouse moves.
    //stage.addEventListener(MouseEvent.CLICK, captureAllClicks);//capture all clicks.
    tocMenuShow();
    // load first movie using the bookmark to find the index
    startVideo(this[_items[0].@instanceName]);// start by playing the first video. revise to call t
}

// ------------- Identity buttons----------------------
function identityInitialize():void {
    //trace("intializing Identity buttons");
    sw_mc.addEventListener( MouseEvent.CLICK, identityClicked );
    sw_mc.buttonMode=true;
    sw_mc.mouseChildren=false;
    sw_mc.addEventListener( MouseEvent.MOUSE_OVER, identityHover );
    identityHoverOffAction();
}
function identityHover(event:MouseEvent):void {
    identityHoverAction();
    event.target.addEventListener( MouseEvent.MOUSE_OUT, identityHoverOff );
}
function identityHoverAction():void {
    Tweener.removeTweens(identityHover_mc);
    Tweener.addTween(identityHover_mc, {alpha:1, time:.25, delay:0, transition:"linear"});
}
function identityHoverOff(event:MouseEvent):void {
    identityHoverOffAction();
    event.target.removeEventListener( MouseEvent.MOUSE_OUT, identityHoverOff );
}
function identityHoverOffAction():void {
    Tweener.removeTweens(identityHover_mc);
    Tweener.addTween(identityHover_mc, {alpha:0, time:3, delay:3, transition:"easeOutExpo"});
}
function identityClicked(event:MouseEvent):void {
    //trace("identityClicked");
    var mailMsg:URLRequest=new URLRequest("mailto:design@SteveWhetstone.com");
    var variables:URLVariables = new URLVariables();
    variables.subject="website response";
    variables.body="I saw your website and ";
    mailMsg.data=variables;
    mailMsg.method=URLRequestMethod.GET;
    navigateToURL(mailMsg);
}

// ------------- toc (table of contents) buttons ---------------------

var tocFormat = new TextFormat();
//tocFormat.kerning = true;// doesn't seem to do much
tocFormat.letterSpacing=1;

function tocInitialize():void {
    // first set the custom message for the first button
    var _customMessage:XMLList;
```

```
    _customMessage=xmlData.message;
    toc00_mc.customMessage_txt.text=_customMessage[0].@messageText;
    // next make all the buttons active and fill in their titles.
    for (i = 0; i < _items.length(); i++) {
        trace("_items[i].@videoURL="+_items[i].@videoURL);
        this[_items[i].@instanceName].addEventListener( MouseEvent.CLICK, tocClicked );
        this[_items[i].@instanceName].buttonMode=true;
        this[_items[i].@instanceName].mouseChildren=false;
        this[_items[i].@instanceName].addEventListener( MouseEvent.MOUSE_OVER, tocHover );
        // maybe use below later to set the label in flash.  but since it's kerning is weak use the
        this[_items[i].@instanceName].displayName_txt.defaultTextFormat=tocFormat;
        this[_items[i].@instanceName].displayName_txt.text=_items[i].@displayName;
        //this[_items[i].@instanceName].displayName_txt.text="";// delete the filler text.

        this[_items[i].@instanceName].videoURL=_items[i].@videoURL;
        this[_items[i].@instanceName].actionText=_items[i].@actionText;
        this[_items[i].@instanceName].actionResult=_items[i].@actionResult;
    }
}
function tocMoveCurrentIndicator(tocTarget:Object) {
    Tweener.addTween(navBulletCurrent_mc, {x:tocTarget.x, y:tocTarget.y,time:.5, transition:"easeOu
    bookMark=replaceSpaces(tocTarget.displayName_txt.text);
    var windowTitle:String="SteveWhestone.com - Portfolio - "+tocTarget.displayName_txt.text;
    placeBookmark(bookMark, windowTitle);
    // add time and click location to database for user logging usuabilty analysis.
}
function tocHover(event:MouseEvent):void {
    Tweener.removeTweens(event.target.navBulletOver_mc);
    Tweener.addTween(event.target.navBulletOver_mc, {alpha:1, time:0, delay:0, transition:"linear"}
    event.target.addEventListener( MouseEvent.MOUSE_OUT, tocHoverOff );
}
function tocHoverOff(event:MouseEvent):void {
    Tweener.removeTweens(event.target.navBulletOver_mc);
    Tweener.addTween(event.target.navBulletOver_mc, {alpha:0, time:1, delay:0, transition:"easeOutE
    event.target.removeEventListener( MouseEvent.MOUSE_OUT, tocHoverOff );
}
function tocClicked(event:MouseEvent):void {
    var now:Date = new Date();
    recordClick(event.target.displayName_txt.text);
    lastClickTime=now;// use to set time out on capture mouse moves in case of lazy click.
    startVideo(event.target);
    hideMouse();
}

// ------------- forward and back buttons  ----------------------
backward_mc.addEventListener( MouseEvent.CLICK, backClick);
backward_mc.addEventListener( MouseEvent.MOUSE_OVER, backHover );
backward_mc.addEventListener( MouseEvent.MOUSE_DOWN, forwardBackDown );
backward_mc.buttonMode=true;
backward_mc.mouseChildren=false;

forward_mc.addEventListener( MouseEvent.CLICK, forwardClick);
forward_mc.addEventListener( MouseEvent.MOUSE_OVER, forwardHover );
forward_mc.addEventListener( MouseEvent.MOUSE_DOWN, forwardBackDown );
```

```
forward_mc.buttonMode=true;
forward_mc.mouseChildren=false;

function forwardBackDown(event:MouseEvent):void {
    //trace("forwardBackDown");
    forwardBackDown_mc.alpha=1;
    event.target.addEventListener( MouseEvent.MOUSE_UP, forwardBackUp );
    event.target.addEventListener( MouseEvent.MOUSE_OUT, forwardBackUp );
}
function forwardBackUp(event:MouseEvent):void {
    //trace("forwardBackUp");
    forwardBackDown_mc.alpha=0;
    event.target.removeEventListener( MouseEvent.MOUSE_UP, forwardBackUp );
    event.target.removeEventListener( MouseEvent.MOUSE_OUT, forwardBackUp );
}

function forwardClick(event:MouseEvent):void {
    //trace("forwardClick");
    playNext();
}
function forwardHover(event:MouseEvent):void {
    //trace("forwardHover");
    forwardHover_mc.alpha=1;
    event.target.addEventListener( MouseEvent.MOUSE_OUT, forwardHoverOff );
}
function forwardHoverOff(event:MouseEvent):void {
    //trace("backHoverOff");
    forwardHover_mc.alpha=0;
    event.target.removeEventListener( MouseEvent.MOUSE_OUT, forwardHoverOff );
}

function backClick(event:MouseEvent):void {
    //trace("backClick");
    playPrev();
}
function backHover(event:MouseEvent):void {
    //trace("backHover");
    backHover_mc.alpha=1;
    event.target.addEventListener( MouseEvent.MOUSE_OUT, backHoverOff );
}
function backHoverOff(event:MouseEvent):void {
    //trace("backHoverOff");
    backHover_mc.alpha=0;
    event.target.removeEventListener( MouseEvent.MOUSE_OUT, backHoverOff );
}

// ------------- hiding and showing toc menu related things--------------------
var tocMenuShowing:Boolean=true;
var manualPaused:Boolean=false;
var lastClickTime:Date = new Date();
var lastAutoPlayTime:Date = new Date();
var autoPlayMinimum:int=3000;// minimum time to show the menu between autoplay playlist videos in m

function hideMouse():void {
```

```
    Mouse.hide();
}
function showMouse():void {
    Mouse.show();
}
function hideSpinner() {
    navBulletCurrent_mc.loadingSpinner_mc.gotoAndStop("spinHide");
}
function showSpinner() {
    navBulletCurrent_mc.loadingSpinner_mc.gotoAndPlay("spinPlay");
    //tocVideoMenuBackground.pause();

}

function tocAlpha(newAlpha:int):void {
    Tweener.removeTweens(navBulletCurrent_mc.alpha);
    Tweener.addTween(navBulletCurrent_mc, {alpha:newAlpha, time:0.25, transition:"easeInExpo"});
    //tocVideoMenuBackground.alpha=newAlpha;
    toc_image_background_mc.alpha=newAlpha;
    for (i = 0; i < _items.length(); i++) {
        Tweener.removeTweens(this[_items[i].@instanceName]);
        Tweener.addTween(this[_items[i].@instanceName], {alpha:newAlpha, time:0.25, delay:0, transi
    }
}
function tocMenuHide():void {
    //trace("tocMenuHide");
    if (tocMenuShowing) {
        tocMenuShowing=false;
        Tweener.removeTweens(callToAction_mc);
        Tweener.addTween(callToAction_mc, {alpha:1, time:0.25, delay:0, transition:"linear"});
        tocAlpha(0);
        //tocVideoMenuBackground.pause();
        if (! manualPaused) {
            myPlayer.play();
        }
    }
}
function tocMenuShow():void {
    //trace("tocMenuShow");
    if (! tocMenuShowing) {
        tocMenuShowing=true;
        showMouse();// mouse may have been hidden if last action was clicking on toc button.
        Tweener.removeTweens(callToAction_mc);
        Tweener.addTween(callToAction_mc, {alpha:0, time:1, delay:2, transition:"linear"});
        tocAlpha(1);
        //tocVideoMenuBackground.source="video/5secondmenuloop-desktop.m4v";
        //tocVideoMenuBackground.play();
        if (myPlayer.paused) {
            //trace("player is already paused");
            manualPaused=true;
        } else {
            manualPaused=false;
            myPlayer.pause();
        }
```

```
    }
}

var mouseOldX:int=0;
var mouseOldY:int=0;
function captureMove(event:MouseEvent):void {
    var minimumMove:int=1;
    var mouseMove:int=Math.abs(mouseOldX-event.stageX)+Math.abs(mouseOldY-event.stageY);
    // trace("mouseMove = "+mouseMove);
    if (mouseMove>minimumMove) {
        mouseOldX=event.stageX;
        mouseOldY=event.stageY;
        if ((80 < event.stageY) && (event.stageY < 560) && (45 < event.stageX) && (event.stageX < 9
            //trace("mouse move in nav box");
            var now:Date = new Date();
            if ((now.time - lastClickTime.time) > 2000) {
                tocMenuShow();
                showMouse();
            } else {
                //trace("mouse moved but it's so soon after a click it was probably a sloppy click
            }
        } else {
            //trace("mouse move not in nav box");
            tocMenuHide();
            showMouse();
        }
    }
}

// ------------- video control functions----------------------
function startVideo(target:Object):void {
    if (myPlayer.source==target.videoURL) {
        //trace("loaded same movie");
        hideSpinner();
    } else {
        //trace("loaded new movie");
        recordViewing(target.displayName_txt.text);
        showSpinner();
    }

    myPlayer.source=target.videoURL;
    myPlayer.play();

    tocMoveCurrentIndicator(target);
    Tweener.removeTweens(target.navVisited01_mc);
    Tweener.addTween(target.navVisited01_mc, {alpha:1, time:.5, delay:0, transition:"linear"});
    callToAction_mc.callToAction_txt.text=target.actionText;
    if (target.actionResult=="") {
        //trace("actionResult is empty and blank");
    } else {
        //trace("actionResult is valid url");
        var callToActionFormat = new TextFormat();
        callToActionFormat.url=target.actionResult;
        callToActionFormat.letterSpacing=1;
```

```
        callToActionFormat.underline=true;
        callToActionFormat.align="right";
        callToAction_mc.callToAction_txt.setTextFormat(callToActionFormat);
        callToAction_mc.callToAction_txt.width=callToAction_mc.callToAction_txt.textWidth+20;
        callToAction_mc.callToAction_txt.x=0-callToAction_mc.callToAction_txt.width;
    }
}

function movieEnd(event:VideoEvent) {
    //trace("Detected end of movie");
    var now:Date = new Date();
    // userTrack( now,event.target.videoURL);// record this in database for user analysis and feedb
    lastAutoPlayTime=now;// use to set time out on capture mouse moves in case of lazy click.
    if (autoPlaysLeft>0){
        autoPlaysLeft--;
        trace("autoPlaysLeft="+autoPlaysLeft);
    playNext();// playing the first video. revise to call the playlist function.
    }
}

function playNext() {
    var playlistIndex:int=0;// use to track what to play next
    tocMenuShow();
    showSpinner();
    playlistIndex=bookmarkIndex(bookMark)+1;// the next item in the xml list is up.
    if (playlistIndex>=_items.length()) {
        //trace("reset playlist index");
        playlistIndex=0;// this will set the playlist to the beginging/first entry
    }
    startVideo(this[_items[playlistIndex].@instanceName]);
}
function playPrev() {
    var playlistIndex:int=0;// use to track what to play next
    tocMenuShow();
    showSpinner();
    playlistIndex=bookmarkIndex(bookMark)-1;// the next item in the xml list is up.
    if (playlistIndex<0) {
        //trace("reset playlist index");
        playlistIndex=0;
    }
    startVideo(this[_items[playlistIndex].@instanceName]);// playing the first video. revise to cal
}
function tocVideoEnd(event:VideoEvent) {
    // trace("Detected end of movie");
    //tocVideoMenuBackground.seek(0);
    //tocVideoMenuBackground.play();
}
function playingState(event:VideoEvent) {
    // playlist loads too fast for the user to see menu. pause video here so user gets to see menu
    var now:Date = new Date();
    if ((now.time - lastAutoPlayTime.time) < 2000) {
        //trace("less than 2 seconds since autoplay so add a delay to let the user see the menu");
        var autoPlayTimer:Timer=new Timer(autoPlayMinimum-now.time+lastAutoPlayTime.time,1);
        autoPlayTimer.addEventListener(TimerEvent.TIMER_COMPLETE, autoPlayTimerComplete);
```

```
        autoPlayTimer.start();
        myPlayer.pause();
    } else {
        //trace("ok to play as soon as possible");
        tocMenuHide();
        hideSpinner();
        if ((80 < mouseOldY) && (mouseOldY < 560) && (45 < mouseOldX) && (mouseOldX < 900)) {
            hideMouse();
        }
    }
}
function autoPlayTimerComplete(event:TimerEvent):void {
    myPlayer.play();
    tocMenuHide();
    hideSpinner();
    if ((80 < mouseOldY) && (mouseOldY < 560) && (45 < mouseOldX) && (mouseOldX < 900)) {
        hideMouse();
    }
}

// ------------- bookmark functions ----------------------
function replaceSpaces(str:String):String {
    str=str.toLowerCase();
    var myPattern:RegExp=/ /g;
    return (str.replace(myPattern, "_"));
}

function bookmarkIndex(bookMarkName:String):int {
    // get the current current index in the _items list. if bookmark is not found it returns zero
    var myPlaylistIndex:int=0;
    for (i = 0; i < _items.length(); i++) {
        // loop through _items.displayName_txt.text till it matches the bookmark then play the next
        if (replaceSpaces(_items[i].@displayName)==replaceSpaces(bookMark)) {
            myPlaylistIndex=i;
            //trace("bookmark match found");
        }
    }
    return myPlaylistIndex;
}
function placeBookmark(bookmarkName, windowTitle) {
    // changes title of html window and adds a bookmark to the url for later returning to this page
    if (ExternalInterface.available) {
        // placeBookmark is a javascript function in the html page
        ExternalInterface.call("placeBookmark", bookmarkName, windowTitle);
    }
}

// ------------- fullscreen functions ----------------------
myPlayer.fullScreenTakeOver=false;
//tocVideoMenuBackground.fullScreenTakeOver=false;
function fullScreenHandler( event:FullScreenEvent ) {
    //trace("fullScreenHandler");
    // fixes a bug on macs that makes the user click once to set the flash window in focus and then
    if (stage.displayState==StageDisplayState.NORMAL) {
```

```
        if (ExternalInterface.available) {
            ExternalInterface.call("changeFocus");// this is a javascript function in the html page
        }
    }
    if (stage.displayState==StageDisplayState.FULL_SCREEN) {
        hideMouse();// because now the mouse is probably over the movie. . . annoying.
    }
}
stage.addEventListener( FullScreenEvent.FULL_SCREEN, fullScreenHandler );

// ------------- reporting functions ----------------------

function sendMail(message:String):void {
    var variables:URLVariables = new URLVariables();
    var varSend:URLRequest=new URLRequest("http://www.stevewhetstone.com/feedback/HiQFormMail/HiQFM
    var varLoader:URLLoader=new URLLoader  ;
    varSend.method=URLRequestMethod.POST;
    varSend.data=variables;

    variables.description=message;
    varLoader.load(varSend);
}

function reportUsage(message:String):void {
    var now:Date = new Date();
    //sendMail(message);
}
function recordViewing(movieName:String):void {
    var now:Date = new Date();
    //trace("put an entry in the database with playListPath = " + playListPath + " date = " + now +
}
function recordClick(manualClick:String):void {
    var now:Date = new Date();
    //trace("put an entry in the database with playListPath = " + playListPath + " date = " + now +
}
```