```
// copyright stevewhetstone.com All rights resereved.
import caurina.transitions.Tweener;
stop();

var i:uint;// reusable loop variable
var playListPath:String="index.xml";
var bookMark:String="1";
var recordOn:Boolean;
var playlist:String;// don't assign this to a value here because it resets playlist on return to pa
ge 1 and breaks user tracking by playlist.
var previousPageNumber:Number;
var intitializedYet:Boolean;
var totalTics:Number = 82;

// ------------- Initialize functions ----------------------

function initializeSite() {
    if (! intitializedYet) {
        // this is sloppy code to make sure some things only run once and relies on "initializedYet
g defaulted to false on creation
        intitializedYet=true;
        // it also relies on "initializedYet" not being reset to false when reloading this frame
        previousPageNumber=82;
        makeCat("cat1_mc", 1);
        makeCat("cat2_mc", 4);
        makeCat("cat3_mc", 23);
        makeCat("cat4_mc", 35);
        makeCat("cat5_mc", 47);
        makeCat("cat6_mc", 60);
        makeCat("cat7_mc", 74);
        makeCat("cat8_mc", 79);
        var totalTics:Number=82;
        makeAllTics(totalTics);
    }
    changePage(1, "---------------------------------- by email link");

}


this.loaderInfo.addEventListener(Event.COMPLETE, loaderComplete);// used to check for bookmarks and
 playlist query strings after this swf is loaded.
function loaderComplete(myEvent:Event) {
    if (ExternalInterface.available) {
        var flashVars=this.loaderInfo.parameters;
        if (flashVars.playlist) {
            playlist=flashVars.playlist;
            //playListPath="custom/"+flashVars.playlist+".xml";// not used now, but later should al
mable portfolio pages.
            recordOn=true;
            recordOpening("recieved visit by user "+ playlist +". Detailed report at http://stevewh
om/web/recorder_play_html.php?playlist="+playlist+" ");
        }
        if (flashVars.bookmark) {
            bookMark=flashVars.bookmark;
        }
    }
    //loader.load(new URLRequest(playListPath));// not implemented yet need to revise pages dynamic
y based on xml. . .
```

```
    initializeSite();
}


// ------------ page changing functions---------------------

function moveAllTics(distance: Number) {
    for (var i:uint=1; i <= totalTics; i++) {
        var ticName:String="tic"+i+"_mc";
        this[ticName].y=this[ticName].y+distance;
    }
    /*
    // this code is not neaded. button hovers are moved with timeline keyframes.
    var totalCategories:uint=8;
    for (var j:uint=1; j <= totalCategories; j++) {
        var catName:String="cat"+j+"_mc";
        this[catName].y=this[catName].y+distance;
    }
    */
}

function changePage(newPageNumber:Number, messageForRecording:String):void {
    // hide old page tic
    var newTicName:String="tic"+newPageNumber+"_mc";
    this[newTicName].alpha=1;
    // show new page tic
    var previousTicName:String="tic"+previousPageNumber+"_mc";
    this[previousTicName].alpha=0;
    if (newPageNumber==1) {
        //trace("move tics off stage or disable them or make them invisible");
        moveAllTics(-400);
        //totalTics
    }
    if (previousPageNumber==1) {
        //trace("return tics to stage or enable them or make them visible");
        moveAllTics(400);
    }
    trace("previousPageNumber="+previousPageNumber);
    if (23 < previousPageNumber && previousPageNumber < 35) {
        trace("better stop netstreem");

    }
    previousPageNumber=newPageNumber;
    gotoAndStop(newPageNumber);
    record_add(playlist, newPageNumber, messageForRecording);

}



// ------------------------------------------------------------- arrow buttons-----------------
----
previous_mc.addEventListener( MouseEvent.CLICK, previousClicked );
previous_mc.buttonMode=true;
previous_mc.mouseChildren=false;
previous_mc.addEventListener( MouseEvent.MOUSE_OVER, previousHover );
function previousHover(event:MouseEvent):void {
    event.target.addEventListener( MouseEvent.MOUSE_OUT, previousHoverOff );
```

```
}
function previousHoverOff(event:MouseEvent):void {
    event.target.removeEventListener( MouseEvent.MOUSE_OUT, previousHoverOff );
}
function previousClicked(event:MouseEvent):void {
    changePage(currentFrame-1, "by arrow previous");
    //this.prevFrame();
}

next_mc.addEventListener( MouseEvent.CLICK, nextClicked );
next_mc.buttonMode=true;
next_mc.mouseChildren=false;
next_mc.addEventListener( MouseEvent.MOUSE_OVER, nextHover );
function nextHover(event:MouseEvent):void {
    event.target.addEventListener( MouseEvent.MOUSE_OUT, nextHoverOff );
}
function nextHoverOff(event:MouseEvent):void {
    event.target.removeEventListener( MouseEvent.MOUSE_OUT, nextHoverOff );
}
function nextClicked(event:MouseEvent):void {
    changePage(currentFrame+1, "by arrow next");
    //this.nextFrame();
}


// ------------------------------------------------- page loader on "frame actions" layer -------
-------

function traceDisplayList(container:DisplayObjectContainer,
        indentString:String = ""):void {
    var child:DisplayObject;
    for (var i:uint=0; i < container.numChildren; i++) {
        child=container.getChildAt(i);
        trace(indentString, child, child.name);
        if (container.getChildAt(i) is DisplayObjectContainer) {
            traceDisplayList(DisplayObjectContainer(child), indentString + "    ");
        }
    }
}

function closeAllStreams(evt:Event) {
    //myNetStream.close();
    //mySound.close();
    //myNetConnection.close();
    //myLocalConnection.close();
    trace('function closeAllStreams');
}


function configureListeners(dispatcher:IEventDispatcher):void {
            dispatcher.addEventListener(Event.COMPLETE, loadComplete);
            //dispatcher.addEventListener(HTTPStatusEvent.HTTP_STATUS, httpStatusHandler);
            //dispatcher.addEventListener(Event.INIT, initHandler);
            //dispatcher.addEventListener(IOErrorEvent.IO_ERROR, ioErrorHandler);
            //dispatcher.addEventListener(Event.OPEN, openHandler);
            dispatcher.addEventListener(ProgressEvent.PROGRESS, loadProgress);
            //dispatcher.addEventListener(Event.UNLOAD, unLoadHandler);
        }
```

```
function fullPageLoader(ImagePath:String) {
    trace('function fullPageLoader');
    //loadbar_mc.y=616;
    var request:URLRequest=new URLRequest(ImagePath);
    var myFullPageLoader:Loader = new Loader();
    configureListeners(myFullPageLoader.contentLoaderInfo);
    //myFullPageLoader.LoaderInfo.addEventListener(Event.COMPLETE,loadComplete);
    //myFullPageLoader.loaderInfo.addEventListener(ProgressEvent.PROGRESS,loadProgress);
    //myFullPageLoader.loaderInfo.addEventListener(Event.UNLOAD, closeAllStreams);

    myFullPageLoader.load(request);
    /*

    http://help.adobe.com/en_US/AS3LCR/Flash_10.0/flash/display/LoaderInfo.html

    http://www.kirupa.com/forum/showthread.php?t=283920 about how the URLLoader doesn't works so we
and use loader instead.

    try with Loader
    http://www.actionscript.org/forums/showthread.php3?t=227574

    */
    //fullPageContainer_mc.addChildAt(myFullPageLoader,0);// might allow preloading by layer=pagenu
    fullPageContainer_mc.addChild(myFullPageLoader);
    //trace("----------------------------------new load-------------");
    //trace("myFullPageLoader.name="+myFullPageLoader.name);
    //myFullPageLoader.close();// will stop the download thereby saving bandwidth for other objects
 load.
}
function loadProgress(e:ProgressEvent):void {
    trace("function loadProgress");
    /*percentloaded_mc.percentloaded_txt.text=int(10-e.target.percentLoaded/10);*/
    loadbar_mc.scaleX = (100-e.bytesLoaded)/100;
    //trace("progressHandler: bytesLoaded=" + event.bytesLoaded + " bytesTotal=" + event.bytesTotal
}
function loadComplete(e:Event):void {
    trace('function loadComplete');
    //percentloaded_mc.y = -429;
    loadbar_mc.y=-363;

    //traceDisplayList(fullPageContainer_mc);
    //fullPageContainer_mc.removeChildAt(0);
    /*
    ABOVE LINE removes the last loaded image and sends for garbage collection.
    otherwise it would just keep adding objects to the fullPageContainer_mc and taking up more memo
with every page fullPageLoader call.
    A better solution would be to let it stay and use this technique to preload the next page and s
 old pages so they can be reused
    instead of having to reload them every time.
    */
}

// ------------------------------------------------ button initializer and mouseovers for page li
nks------------
function btn_Hover(event:MouseEvent):void {
    //trace("btn_Hover over instance " + event.target.name + " links to frame " + event.target.link
Page );
    event.target.addEventListener( MouseEvent.MOUSE_OUT, btn_HoverOff );
```

```
}
function btn_HoverOff(event:MouseEvent):void {
    //trace("btn_HoverOff");
    event.target.removeEventListener( MouseEvent.MOUSE_OUT, btn_HoverOff );
}
function btn_Clicked(event:MouseEvent):void {
    //trace("btn_Clicked");
    changePage(event.target.linksToPage, "by " + event.target.name);
}
function makeButton( movieClipName:String, linksToPage: Number) {
    trace("index makebutton movieclip name="+movieClipName);
    this[movieClipName].addEventListener( MouseEvent.CLICK, btn_Clicked );
    this[movieClipName].buttonMode=true;
    this[movieClipName].mouseChildren=false;
    this[movieClipName].addEventListener( MouseEvent.MOUSE_OVER, btn_Hover );
    this[movieClipName].linksToPage=linksToPage;

}

// ------------------------------------------------- Tic initializer and mouseovers on "page tics"
 layer------------
function tic_Hover(event:MouseEvent):void {
    event.target.parent.alpha=1;
    event.target.addEventListener( MouseEvent.MOUSE_OUT, tic_HoverOff );
}
function tic_HoverOff(event:MouseEvent):void {
    if (event.target.linksToPage!=currentFrame) {
        event.target.parent.alpha=0;
    }
    event.target.removeEventListener( MouseEvent.MOUSE_OUT, tic_HoverOff );
}
function tic_Clicked(event:MouseEvent):void {
    changePage(event.target.linksToPage, "by page tic");
}
function makeTic( movieClipName:String, linksToPage: Number) {
    this[movieClipName].page_number_txt.text=String(linksToPage);
    this[movieClipName].hit_mc.addEventListener( MouseEvent.CLICK, tic_Clicked );
    this[movieClipName].hit_mc.buttonMode=true;
    this[movieClipName].hit_mc.mouseChildren=false;
    this[movieClipName].hit_mc.addEventListener( MouseEvent.MOUSE_OVER, tic_Hover );
    this[movieClipName].hit_mc.linksToPage=linksToPage;
    this[movieClipName].alpha=0;
}
function makeAllTics(pageCount: Number) {
    for (var i:uint=1; i <= pageCount; i++) {
        makeTic("tic"+i+"_mc", i);
    }
}

// ------------------------------------------------- category initializer and mouseovers on "categ
ory buttons" layer------------
/*
var whiteColorTransform:ColorTransform = new ColorTransform();
whiteColorTransform.color = 0xFFFFFF;
var blueColorTransform:ColorTransform = new ColorTransform();
blueColorTransform.color = 0x0099D6;
//this[movieClipName].cat_tics_mc.transform.colorTransform=whiteColorTransform;
*/
```

```
function cat_Hover(event:MouseEvent):void {
    event.target.parent.cat_underscore_mc.alpha=1;
    event.target.addEventListener( MouseEvent.MOUSE_OUT, cat_HoverOff );
}
function cat_HoverOff(event:MouseEvent):void {
    event.target.parent.cat_underscore_mc.alpha=0;
    event.target.removeEventListener( MouseEvent.MOUSE_OUT, cat_HoverOff );
}
function cat_Clicked(event:MouseEvent):void {
    changePage(event.target.linksToPage, "by category header");
}
function makeCat( movieClipName:String, linksToPage: Number) {
    this[movieClipName].hit_mc.addEventListener( MouseEvent.CLICK, cat_Clicked );
    this[movieClipName].hit_mc.addEventListener( MouseEvent.CLICK, cat_Clicked );
    this[movieClipName].hit_mc.buttonMode=true;
    this[movieClipName].hit_mc.mouseChildren=false;
    this[movieClipName].hit_mc.addEventListener( MouseEvent.MOUSE_OVER, cat_Hover );
    this[movieClipName].hit_mc.linksToPage=linksToPage;
    this[movieClipName].cat_underscore_mc.alpha=0;
}
// ------------------------------------------------WWW buttons and mouseovers-----------
function email_Hover(event:MouseEvent):void {
    event.target.addEventListener( MouseEvent.MOUSE_OUT, email_HoverOff );
}
function email_HoverOff(event:MouseEvent):void {
    event.target.removeEventListener( MouseEvent.MOUSE_OUT, email_HoverOff );
}
function email_Clicked(event:MouseEvent):void {
    record_add(playlist,1,"------------------------------ Clicked on e-mail link!!!");
    var mailMsg:URLRequest=new URLRequest(event.target.linksToWeb);
    //var mailMsg:URLRequest=new URLRequest("mailto:ODesign@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);
}
function make_Email_Btn( movieClipName:String, linksToWeb: String) {
    this[movieClipName].addEventListener( MouseEvent.CLICK, email_Clicked );
    this[movieClipName].buttonMode=true;
    this[movieClipName].mouseChildren=false;
    this[movieClipName].addEventListener( MouseEvent.MOUSE_OVER, email_Hover );
    this[movieClipName].linksToWeb=linksToWeb;
}


function btn_Web_Hover(event:MouseEvent):void {
    event.target.addEventListener( MouseEvent.MOUSE_OUT, btn_Web_HoverOff );
}
function btn_Web_HoverOff(event:MouseEvent):void {
    event.target.removeEventListener( MouseEvent.MOUSE_OUT, btn_Web_HoverOff );
}
function btn_Web_Clicked(event:MouseEvent):void {
    var webLink:URLRequest=new URLRequest(event.target.linksToWeb);
    webLink.method=URLRequestMethod.GET;
    navigateToURL(webLink);
}
```

```
function make_Web_Btn( movieClipName:String, linksToWeb: String) {
    this[movieClipName].addEventListener( MouseEvent.CLICK, btn_Web_Clicked );
    this[movieClipName].buttonMode=true;
    this[movieClipName].mouseChildren=false;
    this[movieClipName].addEventListener( MouseEvent.MOUSE_OVER, btn_Web_Hover );
    this[movieClipName].linksToWeb=linksToWeb;

}


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

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

function recordOpening(message:String):void {
    sendMail(message);// this doesn't work on local mamp but doesn't crash either.
}

function record_add(the_playlist:String,the_pg:int,the_activity:String):void {
    trace("record_add function");
    if (recordOn) {
        var variables:URLVariables = new URLVariables();
        var varSend:URLRequest=new URLRequest("web/recorder_add.php");
        var varLoader:URLLoader=new URLLoader  ;
        varSend.method=URLRequestMethod.GET;
        varSend.data=variables;
        variables.playlist=the_playlist;
        variables.pg=the_pg;
        variables.activity=the_activity;
        varLoader.load(varSend);

    }
}
```