

Academic Support Center Student Work Slideshow (ActionScript 3.0) Full Code

```
// import tweener
import caurina.transitions.Tweener;

// delay between slides
const TIMER_DELAY:int = 3000;
// fade time between slides
const FADE_TIME:int = 1;

// reference to the current slider container
var currentContainer:Sprite;
// index of the current slide
var intCurrentSlide:int = -1;
// total slides
var intSlideCount:int;
// timer for switching slides
var slideTimer:Timer;
// slides holder
var sprContainer1:Sprite;
var sprContainer2:Sprite;
// slides loader
var slideLoader:Loader;
// url to slideshow xml
var strXMLPath:String = "resources/work.xml";
// slideshow xml loader
var xmlLoader:URLLoader;
// slideshow xml
var xmlSlideshow:XML;

function init():void {

    // add the border:
    addChildAt(border, 0);

    // create new urlloader for xml file
    xmlLoader = new URLLoader();
    // add listener for complete event
    xmlLoader.addEventListener(Event.COMPLETE, onXMLLoadComplete);
    // load xml file
    xmlLoader.load(new URLRequest(strXMLPath));

    // create new timer with delay from constant
    slideTimer = new Timer(TIMER_DELAY);
    // add event listener for timer event
    slideTimer.addEventListener(TimerEvent.TIMER, switchSlide);

    // create 2 container sprite which will hold the slides and
    // add them to the masked movieclip
    sprContainer1 = new Sprite();
    sprContainer2 = new Sprite();
    mcSlideHolder.addChild(sprContainer1);
    mcSlideHolder.addChild(sprContainer2);

    // keep a reference of the container which is currently
    // in the front
    currentContainer = sprContainer2;

    mcSlideHolder.x = 32;
```

Academic Support Center Student Work Slideshow (ActionScript 3.0) Full Code

```
mcSlideHolder.y = 2;

}

function onXMLLoadComplete(e:Event):void {
    // create new xml with the received data
    xmlSlideshow = new XML(e.target.data);
    // get total slide count
    intSlideCount = xmlSlideshow..image.length();
    // switch the first slide without a delay
    switchSlide(null);
}

function fadeSlideIn(e:Event):void {
    // add loaded slide from slide loader to the
    // current container
    currentContainer.addChild(slideLoader.content);
    // clear preloader text
    mcInfo.lbl_loading.text = "";
    // fade the current container in and start the slide timer
    // when the tween is finished
    Tweener.addTween(currentContainer, {alpha:1, time:FADE_TIME, onComplete:function() {
slideTimer.start(); }});
}

function switchSlide(e:Event):void {
    // check, if the timer is running (needed for the
    // very first switch of the slide)
    if(slideTimer.running)
        slideTimer.stop();

    // check if we have any slides left and increment
    // current slide index
    if(intCurrentSlide + 1 < intSlideCount)
        intCurrentSlide++;
    // if not, start slideshow from beginning
    else
        intCurrentSlide = 0;

    // check which container is currently in the front and
    // assign currentContainer to the one that's in the back with
    // the old slide
    if(currentContainer == sprContainer2)
        currentContainer = sprContainer1;
    else
        currentContainer = sprContainer2;

    // hide the old slide
    currentContainer.alpha = 0;
    // bring the old slide to the front
    mcSlideHolder.swapChildren(sprContainer2, sprContainer1);

    // create a new loader for the slide
    slideLoader = new Loader();
    // add event listener when slide is loaded
    slideLoader.contentLoaderInfo.addEventListener(Event.COMPLETE, fadeSlideIn);
    // add event listener for the progress
```

Academic Support Center Student Work Slideshow (ActionScript 3.0) Full Code

```
        slideLoader.contentLoaderInfo.addEventListener(ProgressEvent.PROGRESS,
showProgress);
        // load the next slide
        slideLoader.load(new URLRequest(xmlSlideshow..image[intCurrentSlide].@src));

        // show description of the next slide
        mcInfo.lbl_description.text = xmlSlideshow..image[intCurrentSlide].@desc;
        // show current slide and total slides
        mcInfo.lbl_count.text = (intCurrentSlide + 1) + " / " + intSlideCount + " Slides";
    }

function showProgress(e:ProgressEvent):void {
    // show percentage of the bytes loaded from the current slide
    mcInfo.lbl_loading.text = "Loading..." + Math.ceil(e.bytesLoaded * 100 /
e.bytesTotal) + "%";
}

// init slideshow
init();
```