

Slide.Show Quick Start Guide

Vertigo Software
December 2007

Contents

Introduction	1
Your first slideshow with Slide.Show	1
Step 1: Embed the control	2
Step 2: Configure the control.....	3
Step 3: Provide data to the control.....	4
Where to go from here?	6
CodePlex	6
Vertigo.....	6

Introduction

This guide has one purpose – to help you build your first slideshow with Slide.Show as quickly as possible. It assumes that your scenario is typical – you have a nice collection of photos that you’d like to display on your personal Web site.

Of course, Slide.Show was designed with flexibility in mind, whether it’s embedding Flickr photos directly in a blog post, displaying a line of products using the corporate colors and a watermark, or even providing a fluid, theater-like viewing experience for high-resolution artwork.

But that’s not what this guide is about. Once you’re familiar with the basic steps and tips below, you’ll be ready to customize, extend, and embed your slideshows anywhere on the Web using the comprehensive documentation available in the Slide.Show User Guide. But first, the basics...

Your first slideshow with Slide.Show

The typical scenario involves the following steps:

- Step 1.** Determine the Web page to host the Slide.Show control
- Step 2.** Define a configuration file with the desired behavior and options for the control
- Step 3.** Define a data file with the album and slide data for the control

Step 1: Embed the control

At a minimum, a Web page hosting the Slide.Show control requires the following:

- An external script tag that references the standard **Silverlight.js** file
- An external script tag that references the *Release* version of the **SlideShow.js** file
- An inline script tag that creates and embeds the Slide.Show control in the page

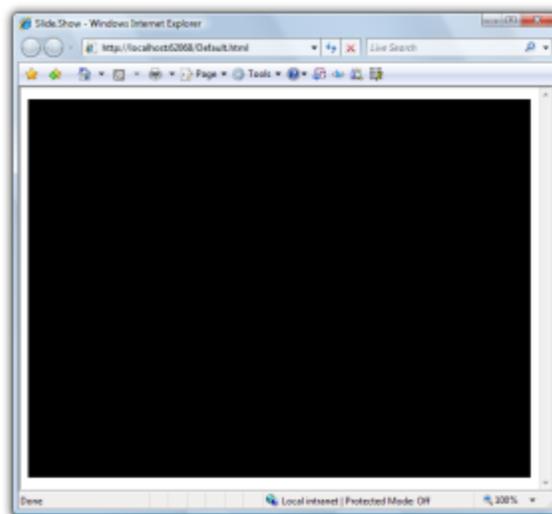
Tip: To keep things organized, create a new folder in your site to contain the page and other files required by your slideshow, including the Silverlight.js and SlideShow.js scripts, configuration and data files, and images.

First, create a new page in your site with the following content:

```
<html>
  <head>
    <title>Slide.Show</title>
    <script type="text/javascript" src="Silverlight.js"></script>
    <script type="text/javascript" src="SlideShow.js"></script>
  </head>
  <body>
    <script type="text/javascript">
      new SlideShow.Control();
    </script>
  </body>
</html>
```

Next, view the page in the browser to make sure everything is wired up correctly. At this point, it's just an empty slideshow as shown below:

Figure 1. An empty slideshow



Note: The inline script tag will embed the Slide.Show control in the HTML document at the exact location of the script tag (e.g. within table cells). This is important to note when integrating the control with complex page layouts.

Step 2: Configure the control

To give substance to your empty slideshow, you'll need to configure it. In the typical scenario, this is done via an XML configuration file.

First, modify the inline script tag in your page as follows:

```
<script type="text/javascript">
    new SlideShow.Control(new SlideShow.XmlConfigProvider());
</script>
```

Note: Slide.Show can be configured via an XML file or JSON (JavaScript Object Notation). The latter allows inline or cross-domain configuration, essential for more complex scenarios (such as embedding the control in a blog post).

Next, create a file named **Configuration.xml** in the *same* folder as your page with the following content:

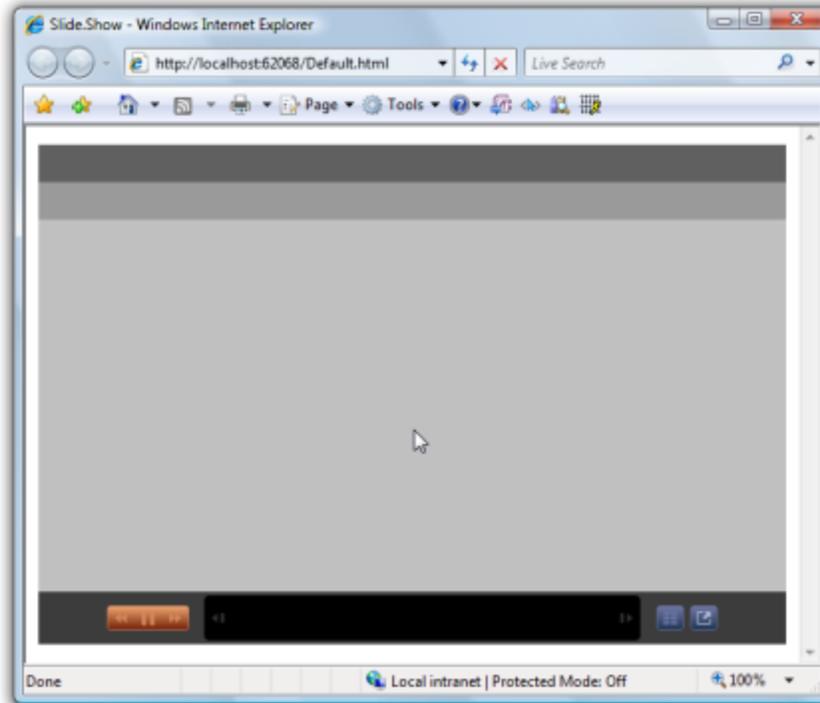
```
<configuration width="600" height="400" background="Silver">
    <modules>
        <module type="SlideViewer" />
        <module type="ProgressBar" />
        <module type="SlideDescription" />
        <module type="NavigationTray">
            <option name="thumbnailViewer.left" value="83" />
        </module>
    </modules>
    <transitions>
        <transition type="FadeTransition" name="CrossFadeTransition" />
        <transition type="WipeTransition" name="WipeRightTransition">
            <option name="direction" value="Right" />
        </transition>
    </transitions>
</configuration>
```

The above configuration overrides the default width, height, and background color of the Slide.Show control, and then adds a few modules and transitions that will provide standard slideshow behavior.

Tip: Slide.Show supports over 300 configurable options! In the above configuration, option tags are used to override the default position of the thumbnail viewer and the default direction of the wipe transition.

Refresh your page in the browser to see the effect of the above configuration – now the control provides both basic and thumbnail navigation, buttons to toggle between album view and full-screen mode, and it will respond to the mouse by sliding in titles and descriptions from the top of the control.

Figure 2. A configured (but still empty) slideshow



Note: You may need to restart the browser or clear the browser's cache to view subsequent changes to the XML configuration file (and to prevent the browser from using a cached version of the file).

Step 3: Provide data to the control

In the previous step, we configured the size, background color, and behavior of the Slide.Show control. In this final step, we'll provide album and slide data to the control via an XML data file.

First, modify the *end* of the configuration file as follows:

```
...
</transitions>
<dataProvider type="XmlDataProvider" />
</configuration>
```

Tip: Alternatively, Slide.Show can be configured to use a different data provider, such as the FlickrDataProvider, which uses the Flickr API to retrieve photo data for a given Flickr user (via REST and JSON). This is useful for more complex scenarios where cross-domain file download restrictions are an issue.

Next, create a file named **Data.xml** in the *same* folder as your page with the following format:

```
<data transition="CrossFadeTransition">
  <album
    title="Album Title #1"
    description="Album Description #1"
    image="Images/Album1.jpg">
    <slide
      title="Slide Title #1"
      description="Slide Description #1"
      image="Images/Slide1.jpg"
      thumbnail="Images/Thumbnail1.jpg" />
    ...
  </album>
  ...
</data>
```

Obviously, your data file should be modified to contain *real* album and slide data specific to your slideshow. Once it's complete, reload the page in the browser and enjoy your slideshow!

Figure 3. The finished slideshow



***Tip:** The data file may reference local or remote images using relative or absolute URLs. JPEG and PNG format is supported.*

***Tip:** Although Slide.Show works well with a single set of images, you can optimize performance and improve the sharpness of smaller images in your slideshow by creating additional sets of images sized specifically for use by albums, thumbnails, and thumbnail previews.*

Where to go from here?

Now that you're familiar with the basic steps and tips above, please refer to the Slide.Show User Guide for comprehensive documentation detailing additional usage scenarios, modules, transitions, configuration and data providers, options, architecture diagrams, contributor guidelines, and tips for extending Slide.Show to meet your unique needs.

CodePlex

Please visit the Slide.Show project page on CodePlex (<http://www.codeplex.com/SlideShow>) for the latest releases, discussions, source code, and to provide feedback, report bugs, suggest new features, or to become an open source contributor for the project.

Vertigo

Please visit the Slide.Show home page (<http://www.vertigo.com/SlideShow.aspx>) to play with a live sample, or for the latest news and additional samples.