Welcome

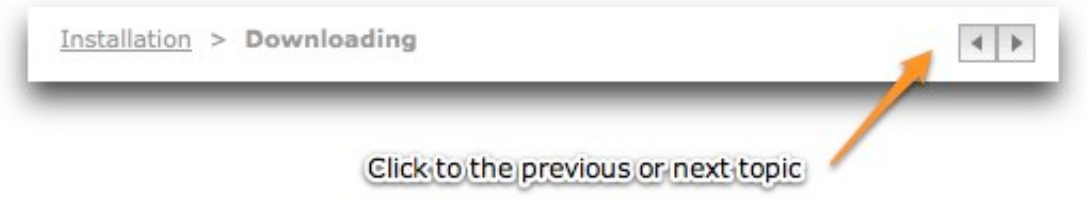
Welcome to the XML Flash Slideshow v4, often referred to as the XFS v4. The Dreamweaver extension allows you to create powerful presentations directly within Dreamweaver, incorporating images, audio, video, and even other SWF animations. The Standalone version allows for the same type of slideshow creation via a standalone application that doesn't require Dreamweaver. The possibilities for customization are endless. While the extensive posibilities are what make the product so special and powerful, it can be a little intimidating when you first get started.

This manual is designed to simplify the usage process, reducing the learning curve and giving instant access to all the information you need. The sections provide basic information to get started, as well as more detailed information and reference which can be used by experienced users for even higher levels of customization and feature integration. As you gain experience with the features and become more acquainted, you'll learn to make even more impressive presentations.

Using the Help

On the left, you'll see the **navigation tree**. This provides hierarchical access to every topic in the documentation. Click on an item to jump directly to it. The current topic is always highlighted. When a topic has child topics (subtopics), this parent topic will often have little real information in itself; instead, it will just provide information about what its subtopics cover. To get to the real meat, you'll need to jump on to its first subtopic.

At the top of the page, above the main page title, there are the **page breadcrumbs** and the **navigation arrows**. The breadcrumbs offer easy links to the parent topics of the current topic. The navigation arrows allow you to to navigate to the previous or next topic:



Click to the previous or next topic

At the top right, you'll see the pdf download link and search box. These allow you to download a printable version of the help files and search the documentation respectively.

To **print an individual topic**, just use the Print command in your browser. The page will be automatically transformed into a printer-friendly format.

Comments

At the bottom of each topic, you can view or add comments relating to that specific topic.   If you wish to disable comments, you can simply add `?&comments=false` to the URL in your browser.

Product Versions

There are two product versions - Basic and Professional. Throughout the manual, those features which are only available in the Pro version are noted. These include:

- **Special Image Displays** - These special image displays offer a display beyond the standard slideshow, including: A 3D Wall/Carousel, a 3D Starfield, an iTunes-style Coverflow, and more.
- **Flickr Photoset Integration Wizard** - This special wizard allows for easy integration of a single Flickr Photoset using the Flickr API. This allows for easy integration beyond the standard Flickr RSS feed.
- **Advanced Color Management** - In the Pro version, you can use an advanced color picker, save swatches and color schemes, or even import a color scheme directly from Adobe's Kuler site. This feature is a big timesaver for serious designers.
- **Flash Component** - The Pro version includes SWC components which can be used to integrate slideshows in existing FLA documents or Flex projects.
- **Popup-Link Builder** - The Pro version includes a tool to quickly build the necessary links to make a customized popup open when an image is clicked.

To upgrade to the Pro version, see the upgrade page.

Ready To Go?

Proceed to the installation section, where you'll learn how to install the XFS v4. Have fun!

QuickStart Guide - Dreamweaver Extension Version

This topic only applies to the Dreamweaver Extension version of the software. If you're using the Standalone version, see this topic.

This topic provides a quick overview of how to get started creating slideshows. **Please note:** All of the topics mentioned in this quickstart are covered in much more detail in the rest of the manual. If you run into trouble or have more questions, view the main manual topics.

Installation - Install Adobe AIR

Unable to detect runtime installation. Please proceed with the AIR installation instructions below.

First, you must download and install the Adobe AIR runtime. Adobe AIR is a free application runtime which allows for the slideshow wizard to run on your system. (To learn more, visit the AIR product page. If you are using Dreamweaver CS4 or higher, AIR will already be installed on your system.) You can download the installer directly by clicking the appropriate link below:

- Windows: AIR Installer for Windows
- Mac: AIR Installer for Mac

If you have any problems with the direct download, you can download and install from Adobe's site:

Get ADOBE® AIR™

Installation - Install the External Wizard

In the `XFS_v4.zip` file you downloaded and uncompressed, double-click the `SSv4_wizard.air` file to launch the Adobe AIR Application Installer and install the external wizard application. Install it to the default location, but disable the 'Launch after Installation' option.

For more information about installing the external wizard, see this topic.

Installation - Install the Dreamweaver Extension

In the `XFS_v4.zip` file you downloaded and uncompressed, double-click the `SSv4_DW_ext.mxp` file to launch the Adobe (Macromedia) Extension Manager and install the Dreamweaver extension. After completing the installation, restart Dreamweaver if it is running.

Note: If double-clicking the file doesn't launch the Extension Manager properly, instead open Dreamweaver and select Help > Manage Extensions. This will launch the Extension Manager; in the Extension Manager window, click the Install button, browse to the `SSv4_DW_ext.mxp` file and click OK.

For more information about installing the Dreamweaver extension, see this topic.

Inserting a New Slideshow

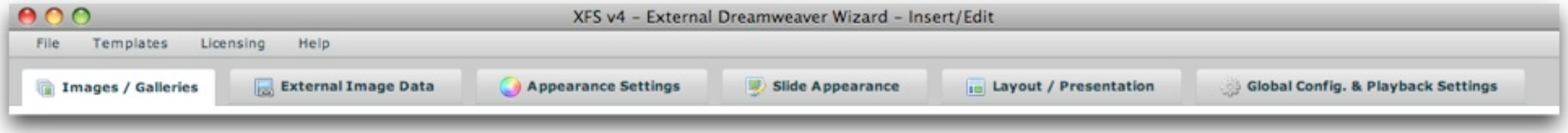
To access the extension, select the *DWUser* section of the Insert Panel (the panel/bar next to the edit window with Common, Layout, Forms, etc. sections; accessed via Window > Insert). On the *DWUser* section, click the *XML Flash Slideshow v4 - Insert Slideshow* button. This will launch the slideshow insertion wizard.

Note: Alternately, you can select Insert > DWUser > XML Flash Slideshow v4 from the Dreamweaver's Insert menu (at the top of the screen, next to File, Edit and View).

For more information about accessing the extension, see this topic. For more information about using the external wizard to create slideshows, see this section.

Basic Usage of Main Slideshow Wizard

When the wizard first launches, you'll be prompted to either create a blank slideshow or start with a starter template. You will then enter the main wizard interface. All of the main wizard sections are accessed via the tabs at the top of the window:

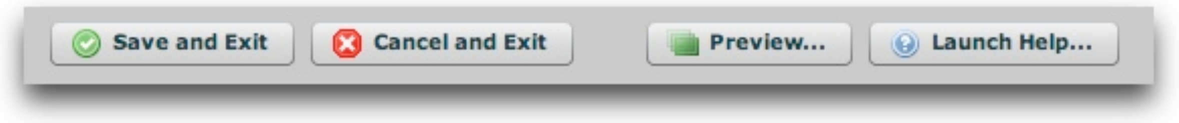


The tabs control the following:

Tab Name	Description
Images / Galleries	This is where you'll add all images to the slideshow, organizing them into one or more galleries.
External Image Data	This is where you can **optionally** specify external data sources such as Flickr feeds, etc.
Appearance Settings	This is where you can control general appearance settings, such as the background color, background image, and preloader

	appearance.
Slide Appearance	This is where you can control the appearance of each of the slides. Options include image scaling settings as well as image matte, border, drop shadow and reflection effects.
Layout / Presentation	This is where you can optionally adjust your slideshow's layout or build a new one from scratch. You can also customize the appearance of all elements included in the user interface.
Global Configuration and Playback Settings	This is where you can specify general slideshow options and playback settings. Options include the image loading mode, automatic playback and the corresponding playback speed, and the image transition configuration.

You can save and exit, cancel and exit, or launch the preview window using the buttons at the bottom of the window:



For more information about using the external wizard to create slideshows, see this section.

Editing a Slideshow After Inserting It

To make changes to a slideshow you've already inserted, select the slideshow in Dreamweaver's Design View. In the Properties panel (accessed via Window > Properties), click the *Edit with Wizard* button.

For more information about accessing the extension to edit a slideshow, see this topic.

Migrating v3 Slideshows to v4

First, its important to note that v3 and v4 operate fully independently -- you can continue to use v3 after installing v4. For information about moving existing v3 slideshows up to v4, see the Migration topic.

QuickStart Guide - Standalone Version

This topic only applies to the Standalone version of the software. If you're using the Dreamweaver Extension version, see this topic.

This topic provides a quick overview of how to get started creating slideshows. **Please note:** All of the topics mentioned in this quickstart are covered in much more detail in the rest of the manual. If you run into trouble or have more questions, view the main manual topics.

Installation - Install Adobe AIR

Unable to detect runtime installation. Please proceed with the AIR installation instructions below.

First, you must download and install the Adobe AIR runtime. Adobe AIR is a free application runtime which allows for the slideshow wizard to run on your system. (To learn more, visit the AIR product page. If you are using Dreamweaver CS4 or higher, AIR will already be installed on your system.) You can download the installer directly by clicking the appropriate link below:

- Windows: AIR Installer for Windows
- Mac: AIR Installer for Mac

If you have any problems with the direct download, you can download and install from Adobe's site:

Get ADOBE® AIR™

Installation - Install the External Wizard

In the `XFS_v4.zip` file you downloaded and uncompressed, double-click the `SSv4_Standalone.air` file to launch the Adobe AIR Application Installer and install the XFS v4 Standalone Slideshow Creator application (also referred to as the "external wizard"). Install it to the default location.

For more information about installing the external wizard, see this topic.

Inserting a New Slideshow

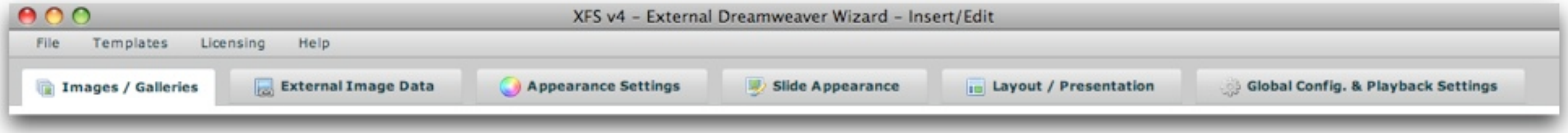
To access the slideshow creator, launch the *XFS v4 Standalone Slideshow Creator* application. There should be a shortcut on your desktop, or you can access it via the Programs / Applications menu.

You'll need to specify the HTML page you want to add a slideshow to, as well as the root folder of the site associated with the page. This folder is the the one that holds all of your sites files, including web pages, images, audio, video and any other assets. For more information about these options and the necessary site configuration to insert a slideshow, see the Getting Started topic.

For more information about accessing the application, see this topic. For more information about using the external wizard to create slideshows, see this section.

Basic Usage of the Standalone Slideshow Creator Wizard

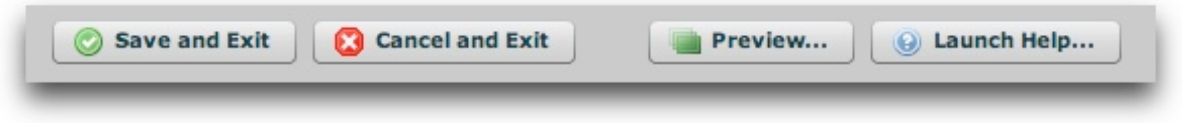
When the wizard first launches, you'll be prompted to either create a blank slideshow or start with a starter template. You will then enter the main wizard interface. All of the main wizard sections are accessed via the tabs at the top of the window:



The tabs control the following:

Tab Name	Description
Images / Galleries	This is where you'll add all images to the slideshow, organizing them into one or more galleries.
External Image Data	This is where you can **optionally** specify external data sources such as Flickr feeds, etc.
Appearance Settings	This is where you can control general appearance settings, such as the background color, background image, and preloader appearance.
Slide Appearance	This is where you can control the appearance of each of the slides. Options include image scaling settings as well as image matte, border, drop shadow and reflection effects.
Layout / Presentation	This is where you can optionally adjust your slideshow's layout or build a new one from scratch. You can also customize the appearance of all elements included in the user interface.
Global Configuration and Playback Settings	This is where you can specify general slideshow options and playback settings. Options include the image loading mode, automatic playback and the corresponding playback speed, and the image transition configuration.

You can save and exit, cancel and exit, or launch the preview window using the buttons at the bottom of the window:



For more information about using the external wizard to create slideshows, see this section.

Editing a Slideshow After Inserting It

To make changes to a slideshow you've already inserted, re-launch the *XFS v4 Standalone Slideshow Creator* application (just like when you initially created the slideshow). When asked for the HTML page and associated site root, enter the same values as those you used when initially creating the slideshow.

For more information about accessing the extension to edit a slideshow, see this topic.

Migrating v3 Slideshows to v4

First, its important to note that v3 and v4 operate fully independently -- you can continue to use v3 after installing v4. For information about moving existing v3 slideshows up to v4, see the Migration topic.

Installation

This section covers how to download and install the extension. Once you've completed this section, you'll be ready to start creating slideshows.

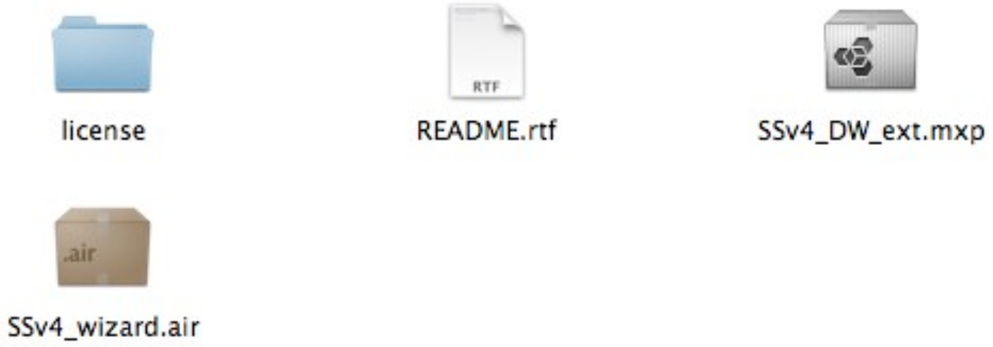
There are four main parts to the installation process which are detailed in each of subtopics:

- Downloading the XFS v4 file
- Installing the Adobe AIR Runtime (used by the external wizard)
- Installing the External Wizard
- Installing the Dreamweaver Extension **(Not applicable for the Standalone version)**

Downloading

Once you've purchased the product, you'll receive a link to download the main ZIP file. This ZIP archive file contains all of the necessary files to install the extension. Note - the file is the same for both the Basic and Pro version; the activated version depends on the serial number you enter. To get started, download the ZIP file and save it to an easy-to-find location such as your desktop. The file will be named `XFS_v4.zip` for the Dreamweaver Extension version, or `XFS_v4_Standalone.zip` for the Standalone version. If you want to find out the build date and build version, you can do so by viewing the `README.rtf` file. After downloading, extract all of the files (unzip the archive).

Once extracted, you should have something similar to the following files:



The `license/` folder contains the end user licensing agreement. The `SSv4_DW_ext.mxp` file is the main Dreamweaver extension (only appears in Dreamweaver Extension version). The `SSv4_wizard.air` file (or `SSv4_Standalone.air` for the Standalone version) is the external wizard application package. The process of installing these components is detailed in the following sections of the manual.

Tip: If you lose your download link, you can access it by locating your purchase confirmation email or by logging into your DWUser.com account.

Download Problems

If, after downloading, you receive a 'corrupted archive' or 'unable to zip' message, this generally indicates that the file was not able to fully download. To address the issue, re-download the file and make sure that the final file size is the same as that listed on the download page. If you try repeatedly and the download still fails, you can try downloading from the alternate site listed on the download page.

Sometimes, firewalls or antivirus software can interfere with the download or shred the file. If you've tried the steps just mentioned and the file still won't open, try temporarily disabling your antivirus and/or firewall.

Installation Prerequisites

Automated Adobe AIR Runtime Detection

Unable to detect runtime installation. Please proceed with the AIR installation instructions below.

Adobe AIR Runtime Installation

Before installing the main extension, you must download and install the Adobe AIR runtime. Adobe AIR is a free application runtime which allows for the slideshow wizard to run on your system. (To learn more, visit the AIR product page.) You can download the installer directly by clicking the appropriate link below:

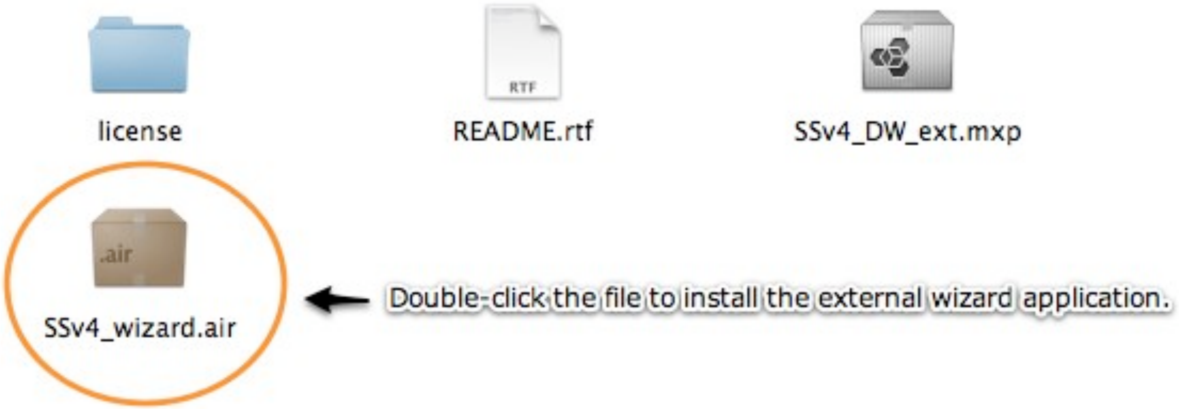
- Windows: AIR Installer for Windows
- Mac: AIR Installer for Mac

If you have any problems with the direct download, you can download and install from Adobe's site:

Get ADOBE® AIR™

Installing the External Wizard

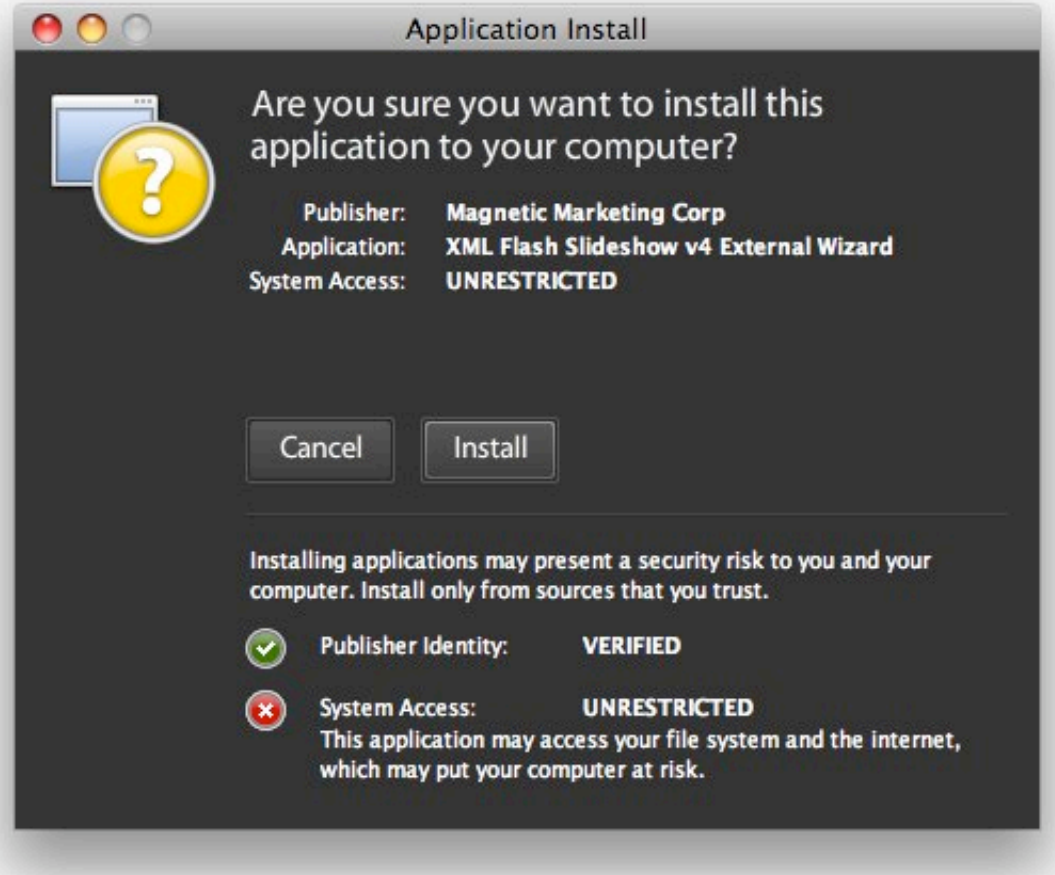
The external wizard is the main application which you will use to create slideshows. It utilizes the Adobe AIR runtime which you just installed (as detailed in the previous section). To install it, locate the `SSv4_wizard.air` file (or `SSv4_Standalone.air` for the Standalone version) in the main ZIP file you downloaded:


license
README.rtf
SSv4_DW_ext.mxp
SSv4_wizard.air
Double-click the file to install the external wizard application.

Double-click the SSv4_wizard.air file (SSv4_Standalone.air in the Standalone version) to install the external wizard. The Adobe Application Installer will launch.

Tip: If you receive an error message about the file type being unrecognized, you can run the application installer directly. To do so, launch `C:\Program Files\Common Files\Adobe AIR\Adobe AIR Application Installer.exe` (Windows) or `/Applications/Utilities/Adobe AIR Application Installer.app` (Mac). You will be prompted to select the installation package; browse to and select the `SSv4_wizard.air` / `SSv4_Standalone.air` file.

Once the installation starts, you will be prompted to confirm the installation. Click **Install** to continue. Note that this demo shows installation of the Dreamweaver Extension version, so the text displayed in some of these dialogs may differ slightly if you're using the Standalone version.



Application Install

Are you sure you want to install this application to your computer?

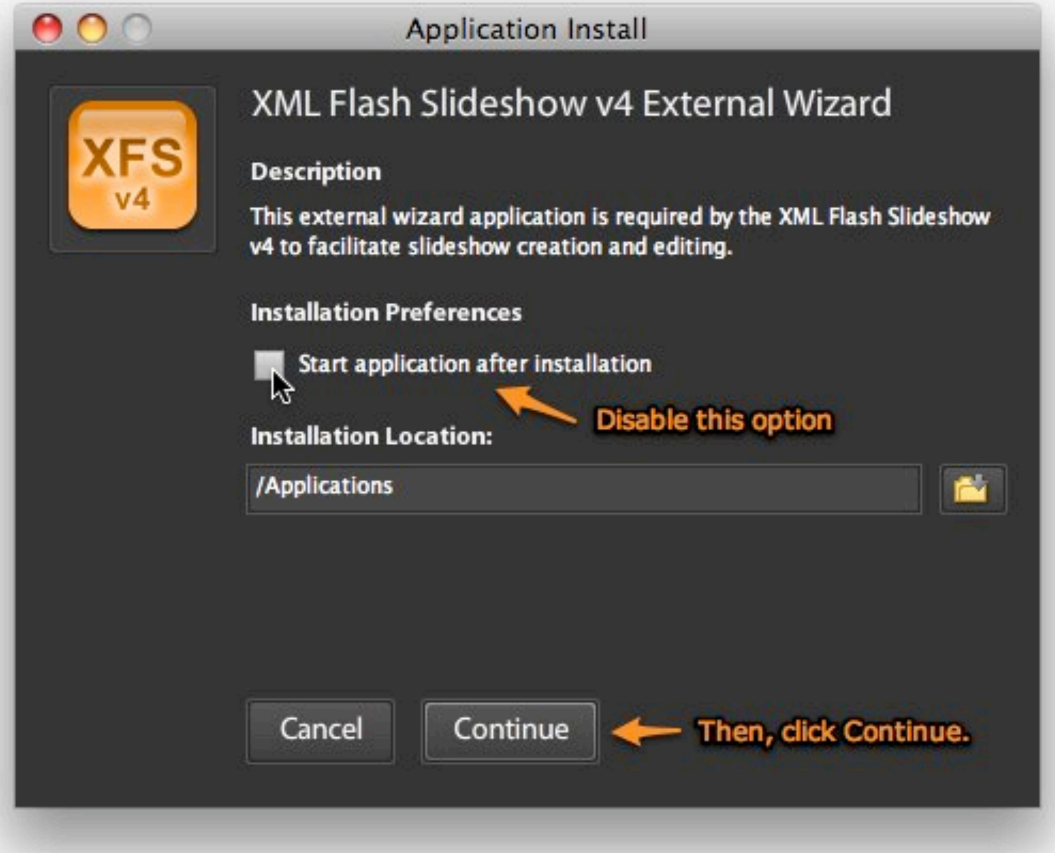
Publisher: Magnetic Marketing Corp
Application: XML Flash Slideshow v4 External Wizard
System Access: UNRESTRICTED

Installing applications may present a security risk to you and your computer. Install only from sources that you trust.

✓ Publisher Identity: VERIFIED

✗ System Access: UNRESTRICTED
This application may access your file system and the internet, which may put your computer at risk.

You will prompted to confirm the installation location. It is generally best to leave the **default value**. (The screenshots are from a Macintosh system, so the default value will vary based on your OS and system configuration). However, if you have installed Dreamweaver to a different drive on your computer, you should install the external wizard to the same folder as Dreamweaver. Disable the '**Start application after installation**' option if you're using the Dreamweaver Extension version, then click **Continue**. If you're using the Standalone version, you can leave it enabled.



Application Install

XML Flash Slideshow v4 External Wizard

Description

This external wizard application is required by the XML Flash Slideshow v4 to facilitate slideshow creation and editing.

Installation Preferences

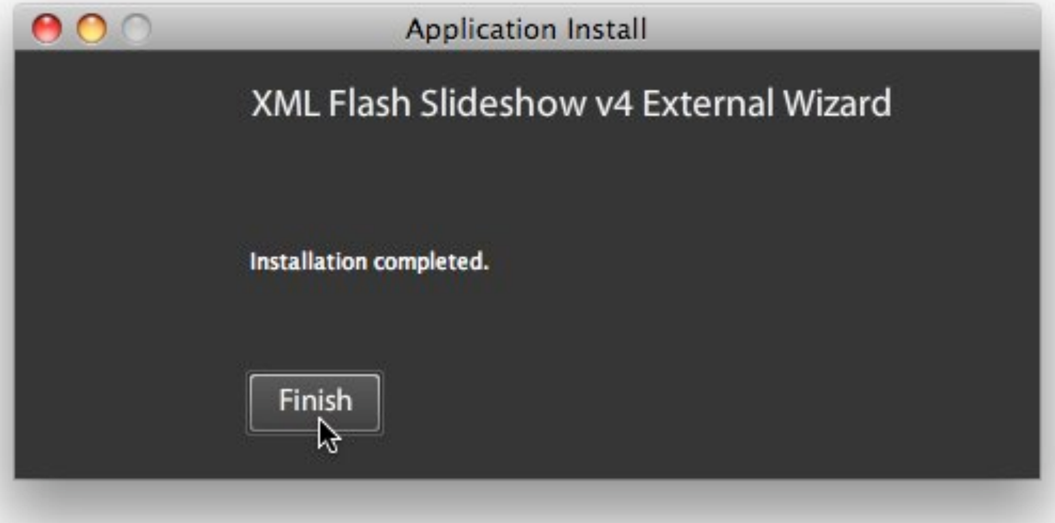
□ Start application after installation

Installation Location:

/Applications

Note for Dreamweaver Extension Version: If you don't see the option to disable 'Start application after installation', the wizard may automatically open after installation. In this case, you will see an error message saying that the wizard can only be launched from within Dreamweaver; just click OK, then proceed with the standard installation.

You will be notified once installation completes. Click **Finish** to exit the installer:



XML Flash Slideshow v4 External Wizard

Installation completed.

The external wizard is now installed, and you're ready to proceed to the next topic.

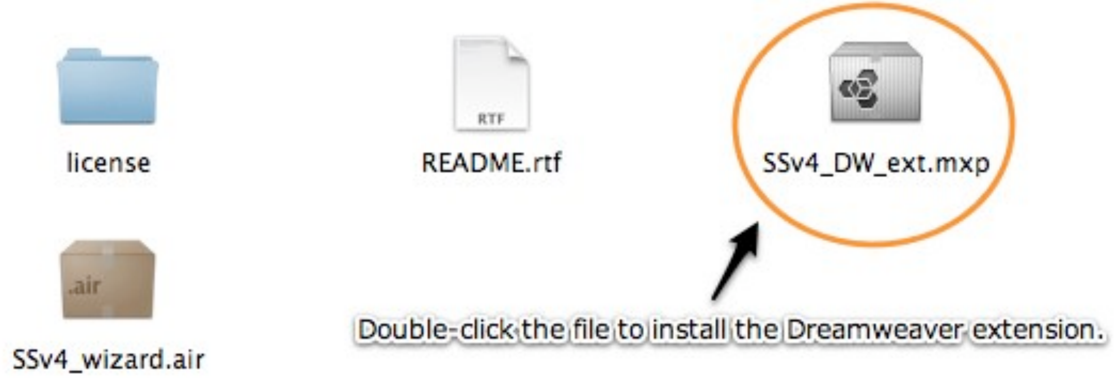
Important: If you haven't installed the Adobe AIR runtime as detained in the previous topic, you won't be able to install the external wizard

Important Note (Dreamweaver Extension Version Only): In the Dreamweaver Extension version, the external wizard is never accessed directly. Instead, it is only accessed from within Dreamweaver. If you attempt to launch the wizard directly, you will receive a warning message indicating that it must be accessed from Dreamweaver.

Installing the Dreamweaver Extension

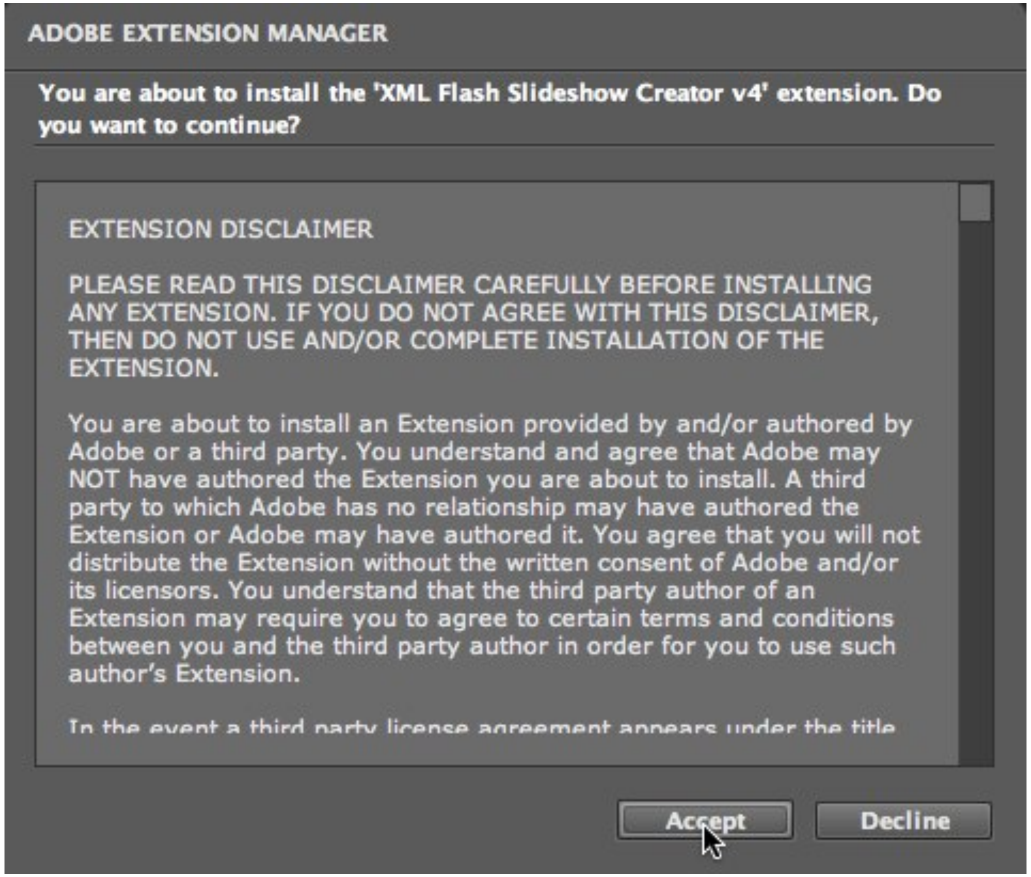
This topic only applies to the Dreamweaver Extension version of the product. Skip this topic if you're working with the Standalone version.

The Dreamweaver Extension is the utility which integrates with Dreamweaver, allowing you to insert slideshows into your webpages. It is required in order to create slideshows. To install it, locate the SSv4_DW_ext.mxp file in the main ZIP file you downloaded:



Tip: If you already have version 2 or 3 of the XML Flash Slideshow installed, you don't have to remove these older versions. All versions operate independently.

The MXP file format is the format used by Adobe Extension Packages. When you installed Dreamweaver or the Adobe Creative Suite, the Adobe Extension Manager (Macromedia Extension Manager for Dreamweaver 8 and prior) was automatically installed. The Extension Manager is configured to automatically open and install MXP extensions. **Double-click the `SSv4_DW_ext.mxp` file** to begin the extension installation. The Extension Manager should open and you will be prompted to accept the extension license. Click **Accept** to install the extension:



ADOBE EXTENSION MANAGER

You are about to install the 'XML Flash Slideshow Creator v4' extension. Do you want to continue?

EXTENSION DISCLAIMER

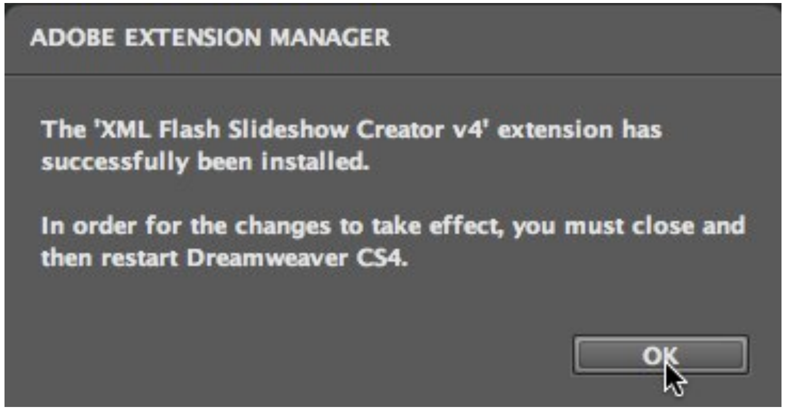
PLEASE READ THIS DISCLAIMER CAREFULLY BEFORE INSTALLING ANY EXTENSION. IF YOU DO NOT AGREE WITH THIS DISCLAIMER, THEN DO NOT USE AND/OR COMPLETE INSTALLATION OF THE EXTENSION.

You are about to install an Extension provided by and/or authored by Adobe or a third party. You understand and agree that Adobe may NOT have authored the Extension you are about to install. A third party to which Adobe has no relationship may have authored the Extension or Adobe may have authored it. You agree that you will not distribute the Extension without the written consent of Adobe and/or its licensors. You understand that the third party author of an Extension may require you to agree to certain terms and conditions between you and the third party author in order for you to use such author's Extension.

In the event a third party license agreement appears under the title

Note: This screenshot is from the Extension Manager CS4. The appearance of older versions of the Extension Manager will differ slightly.

Once the extension installation has completed, you will be notified:



ADOBE EXTENSION MANAGER

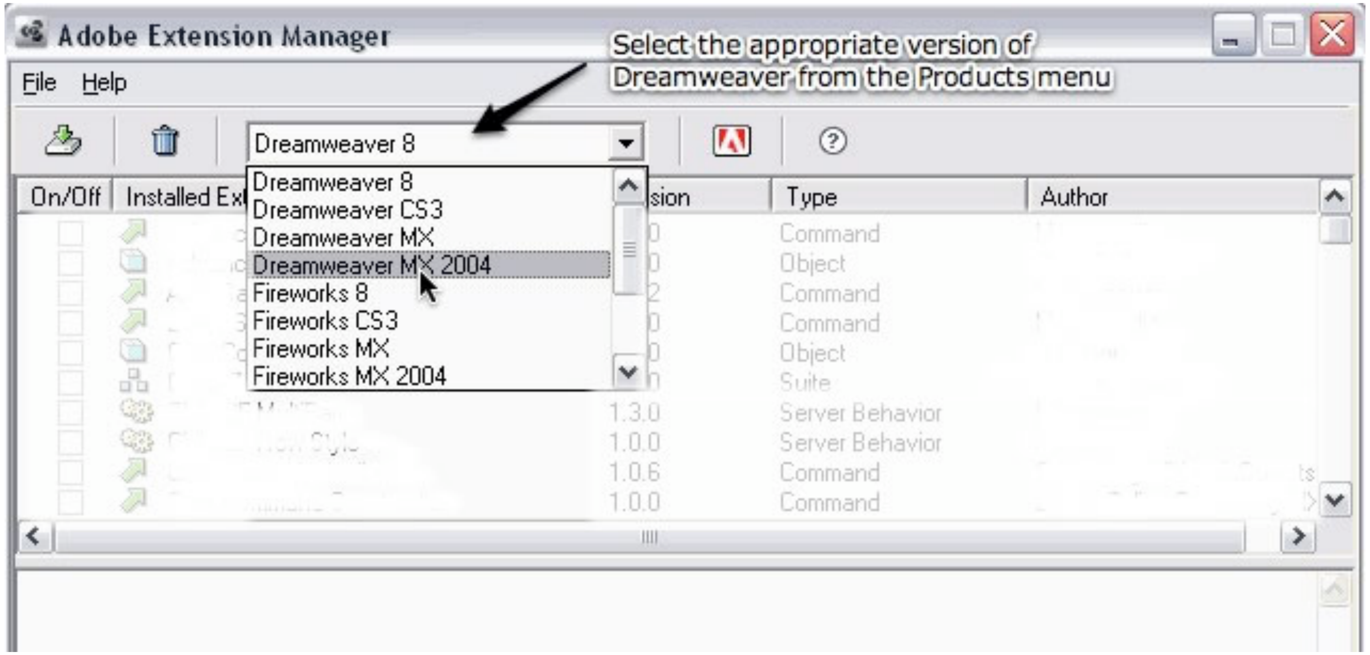
The 'XML Flash Slideshow Creator v4' extension has successfully been installed.

In order for the changes to take effect, you must close and then restart Dreamweaver CS4.

As noted in the message, you must restart Dreamweaver for the extension to be fully installed and ready to use.

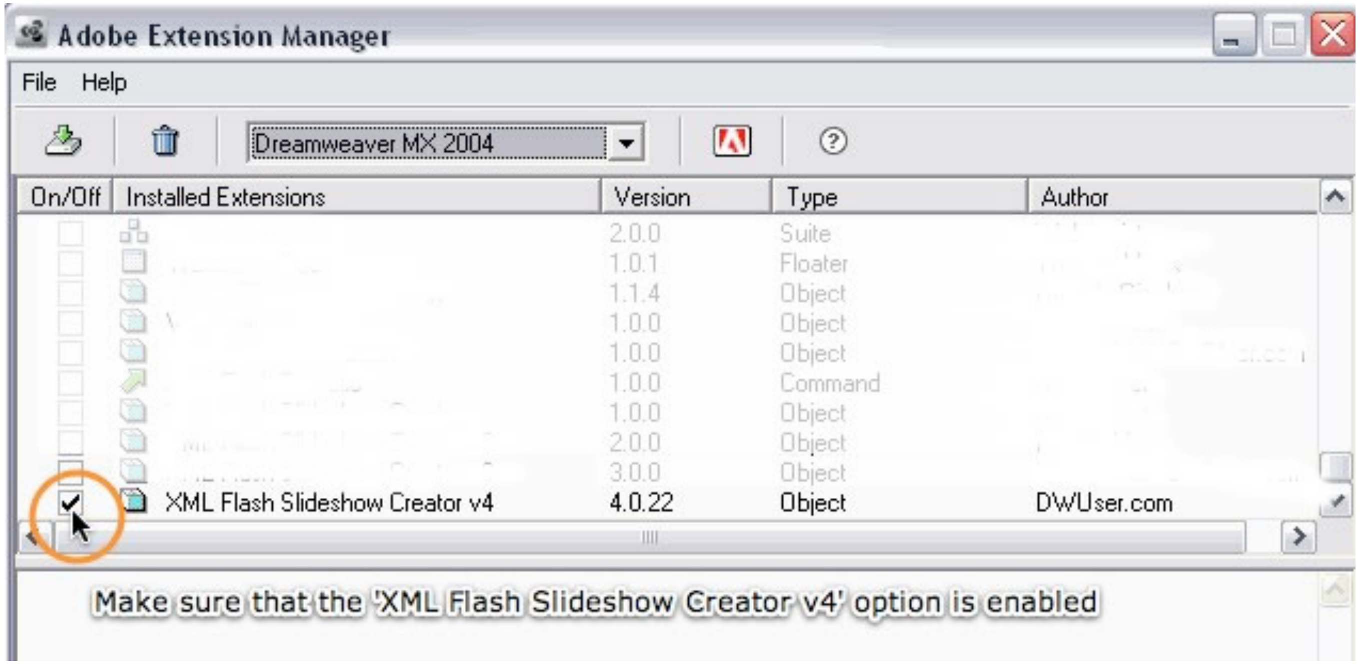
Handling Multiple Dreamweaver Versions

If you have multiple versions of Dreamweaver installed on your system, especially versions prior to CS4, you will need to enable the extension for each of the versions with which you want to use the software. To do this, select each version of Dreamweaver from the products menu individually:



Select the appropriate version of Dreamweaver from the Products menu

Scroll down to the XML Flash Slideshow Creator v4 item, and make sure the **Enabled** checkbox is selected. If not, click it; this will enable the extension for that version of Dreamweaver.



On/Off	Installed Extensions	Version	Type	Author
□		2.0.0	Suite	
□		1.0.1	Floater	
□		1.1.4	Object	
□		1.0.0	Object	
□		1.0.0	Object	
□		1.0.0	Command	
□		1.0.0	Object	
□		2.0.0	Object	
□		3.0.0	Object	
☑	XML Flash Slideshow Creator v4	4.0.22	Object	DWUser.com

Make sure that the 'XML Flash Slideshow Creator v4' option is enabled

Getting Started with the XML Flash Slideshow v4

Now that you have the product installed, you're ready to get started creating slideshows. This portion of the manual gives a high-level overview of the process of using the extension, including workflow and terminology. The topics in this section are covered in more comprehensive detail in subsequent sections. Subtopics include:

- Terms / Vocabulary
- Preparing to Insert a Slideshow
- Workflow for the Dreamweaver Extension Version (How the Extension Works / Integrates with Dreamweaver)
- Workflow for the Standalone Version
- Accessing the Dreamweaver Extension *(only applies to the Dreamweaver extension version)*
- Accessing the Standalone Slideshow Creator Wizard *(only applies to the Standalone version)*
- Previewing Slideshows
- Deploying Slideshows

Getting Started > **Terms / Vocabulary**

Terms / Vocabulary

When using the product, it's important to be familiar with the terminology used. This topic details what each of the various terms used in the wizards and documentation mean.

Term	Meaning
slideshow	This term represents an entire presentation. This is the standalone element which goes in your HTML page and displays images in one or more *galleries*. The examples on our sample page are each a *slideshow*.
presentation	Occasionally used to mean *slideshow* as defined above.
gallery	Within a slideshow, you can have one or more galleries. Each gallery, which can be thought of as an *album*, holds images (slides) related to a specific topic.
album	Occasionally used to mean *gallery* as defined above.
Dreamweaver site	Only applies for Dreamweaver Extension version of the product. When you work in Dreamweaver, you define a *Dreamweaver site* to organize your files. This is simply a definition which tells Dreamweaver where all of the files for a project (website) are located on your computer, and where you're uploading them to (if you're ready to upload files). Think of it as a way to, on your local computer, simulate the setup on your remote web server. The extension requires that you define a Dreamweaver site before inserting a slideshow; this allows it to install the necessary files into your site. To learn about how to define a Dreamweaver site, see this helpful video tutorial on Adobe's site: http://www.adobe.com/designcenter/dreamweaver/articles/lrvid4050_dw.html (very good). Or, see the related page in the Dreamweaver documentation: http://help.adobe.com/en_US/Dreamweaver/10.0_Using/WSc78c5058ca073340dcda9110b1f693f21-7a26.html .
Root Folder of Site	The root folder of your site is the folder that holds all of your site's files, including web pages, photos, audio, video and any other assets. For organization, you must have a file structure of this type, where all site-related files are stored in a single folder (and/or its subfolders) to use the product. This ensures that your site will appear properly when deployed to the web.
XFS v4 Standalone Slideshow Creator	Only applies for Standalone version of the product. In the Standalone version, this is the main wizard application

(AKA Standalone Wizard) *(AKA Standalone Slideshow Creator)*	where slideshow creation and editing takes place. This application can be launched from either your desktop or your Applications / Programs menu.
External Wizard	This is the main wizard application, where slideshow creation and editing takes place. For the Dreamweaver Extension version of the software, this application is launched from within Dreamweaver. For the Standalone version of the software, this is just another name for the XFS v4 Standalone Slideshow Creator application (see previous definition).
XML Configuration File	Each slideshow you create has a unique XML file which defines the slideshow's setup, including layout, appearance. These files also define the image galleries for the slideshow, telling the slideshow where the image, video and audio files should be loaded from.

Preparing to Insert a Slideshow

Needed Preparations: Dreamweaver Version

Nutshell summary (the steps below in a nutshell):

- Create a Dreamweaver Site
- Create and save a new HTML page in your site, or open an existing HTML page in the site. You will insert the slideshow into this page.

First, you must have defined a Dreamweaver site to hold your files. This ensures that all file references can be properly constructed, and the necessary supporting files copied into place. If you're not familiar with what a Dreamweaver site is, you can learn about how to set one up (called "defining a Dreamweaver site") by viewing this helpful video tutorial on Adobe's site: http://www.adobe.com/designcenter/dreamweaver/articles/lrvid4050_dw.html .

Once you have the Dreamweaver site defined, open an existing HTML page within the site or create a new one and save it within the site. It can be any type of HTML page - e.g. PHP, Coldfusion, ASP, .NET, etc; there are no requirements for the page type beyond just being an HTML document. The file must be saved within the site's root folder.

Tip: Some users get confused and try to insert a slideshow into one of the dynamic data template files (accessed by selecting File > New > Page From Sample > XML Flash Slideshow v4. This won't work properly; the dynamic data template files are only for advanced users who are integrated dynamic data sources. Instead, use a normal HTML page.

Needed Preparations: Standalone Version

Nutshell summary (the steps below in a nutshell):

- Have a single folder that holds all of your sites files, folders and assets. This is called your site's "root folder".
- Have an existing HTML page somewhere inside your site's root folder (or one of its subfolders). This can be a page you just created, or an existing one that already has content. You will insert the slideshow into this page.

You must have a single folder that holds all of your site's files, including webpages, images, audio, video and any other assets. This is called your site's "root folder". The page you intend to add the slideshow to must be somewhere within this folder or one of its subfolders. This page can be an existing one that already contains content, or a new one that you create specifically to hold the new slideshow. It can be any type of HTML page - e.g. PHP, Coldfusion, ASP, .NET, etc; there are no requirements for the page type beyond just being an HTML document.

Tip: Some users get confused and try to insert a slideshow into one of the dynamic data template files (accessed by selecting File > New > Page From Sample > XML Flash Slideshow v4. This won't work properly; the dynamic data template files are only for advanced users who are integrated dynamic data sources. Instead, use a normal HTML page.

How the Slideshow Files Work (both versions) + General Tips

Each slideshow has a unique XML configuration file, while sharing a common SWF and javascript file. The javascript file handles all embedding of the SWF and enables the various javascript-to-XFSv4 communication paths. The XML configuration file defines the appearance (layout, colors, etc.) of a slideshow, as well as the image galleries to be displayed. When you insert a slideshow, a block of slideshow HTML is created and inserted into the page. This block is required for the slideshow to be displayed. It embeds the slideshow SWF in the host HTML page, passing all of the configuration information including the path to the XML configuration file and any additional XML data sources. The slideshow HTML also passes parameters which define global appearance settings (such as the background color, opacity, and image) as well as the appearance of the slideshow while the external configuration data is being loaded.

Tip: You cannot view the slideshow SWF directly. Doing so will result in an 'Unable to load XML' error being shown. The slideshow *must* be viewed from within an HTML page, due to the required configuration information which is passed by the slideshow HTML code. Without this information, the slideshow is unable to be displayed.

Tip: When you insert a slideshow, all of the file paths are written to the XML configuration file. They are relative, however, to the HTML page which holds the actual slideshow. Because of this, you cannot simply drag-and-drop or copy-n-paste a slideshow from one page to another unless both pages are in the same directory. Moving a slideshow to a page in a different directory will result in "image failed to load" messages, because the paths will be incorrect.

Tip: Copying a pasting a slideshow will result in two slideshows which reference the same XML configuration file. This will result in edits of the one slideshow affecting the other as well. If you wish to duplicate a slideshow, you should instead re-run the Insert Wizard. You will be given the option to copy the configuration of an existing slideshow when creating the new slideshow.

Workflow: How the Dreamweaver Extension Version Works / Integrates with Dreamweaver

This topic only applies to the Dreamweaver extension version of the product. If you're using the Standalone version, see this topic.

This topic explains how the extension integrates with Dreamweaver, giving a high-level overview of how the slideshow creation workflow functions.

Please Note: The actual process of completing each of these workflow steps is detailed in the subsequent topics; if you're a new user, you can probably skip this topic.

Behind the scenes, the process of creating a slideshow involves several steps:

- If this is the first slideshow in the site, all necessary supporting files are copied into the site and placed in a designated folder
- A new XML configuration file is created for the slideshow
- All configuration information and image gallery information is written to this configuration file
- A block of slideshow HTML is constructed; this contains all of the global configuration settings as well as the path to the XML configuration file
- This slideshow HTML is inserted into your HTML page, where it will behave like any other page element (e.g. it can be dragged-and-dropped anywhere in the page)

The Insert Wizard handles all of these steps automatically for you. To create a slideshow, you'll use the following workflow:

- Open the page you want to insert a slideshow into (it should be prepared as explained in the Preparing to Insert a Slideshow topic).
- Run the Insert Wizard
- The Dreamweaver extension dialog will open. It acts as the liason between the main slideshow creation wizard (the external wizard) and Dreamweaver. It will only be visible for a moment, then the external wizard will open.
- The external wizard will guide you through the creation process. When you're finished, you'll save and exit the wizard. To actually insert the slideshow and return to Dreamweaver, you must then click the Insert button in the Dreamweaver extension dialog.
- The Slideshow HTML code block will be inserted into your HTML page and the XML configuration file created and placed in the supporting files directory.

Tip: To create additional slideshows, you'll just run the Insert Wizard again.

To edit a slideshow, you'll use a similar workflow:

- In Design View, click the slideshow
- In the Property Inspector (Window > Properties), you will see the Slideshow Inspector. Click the **Edit** button to launch the Edit Wizard.
- As when inserting the Dreamweaver extension dialog will momentarily appear, then the external wizard will launch and load in the slideshow in question, allowing you to make changes.
- When you're finished, you'll save changes and exit the external wizard. To actually update your slideshow HTML code and return to Dreamweaver, you must then click the Update button in the Dreamweaver extension dialog.
- The Slideshow HTML and the XML configuration files will be updated based on any configuration changes you made

Workflow: How the Standalone Version Works

This topic only applies to the Standalone version of the product. If you're using the Dreamweaver Extension version, see this topic.

This topic explains how the Standalone Slideshow Creator is used, giving a high-level overview of how the slideshow creation workflow functions.

Please Note: The actual process of completing each of these workflow steps is detailed in the subsequent topics; if you're a new user, you can probably skip this topic.

Behind the scenes, the process of creating a slideshow involves several steps:

- If this is the first slideshow in the site, all necessary supporting files are copied into the site and placed in a designated folder
- A new XML configuration file is created for the slideshow
- All configuration information and image gallery information is written to this configuration file
- A block of slideshow HTML is constructed; this contains all of the global configuration settings as well as the path to the XML configuration file
- This slideshow HTML is inserted into your HTML page, where it will behave like any other page element (e.g. it can be dragged-and-dropped anywhere in the page)

The Standalone Slideshow Creation Wizard handles all of these steps automatically for you. To create a slideshow, you'll use the following workflow:

- Have an existing page you want to insert a slideshow into (it should be prepared as explained in the Preparing to Insert a Slideshow topic). This can be a newly-created page or one with existing content.
- Launch the XFS v4 Standalone Slideshow Creator application
- Select the HTML page you want to add the slideshow to. Also, select the root folder for your the page's associated site.
- The external wizard will guide you through the creation process. When you're finished, you'll save and exit the wizard.
- The slideshow HTML code block will be inserted directly into the top of your HTML page and the XML configuration file created and placed in the supporting files directory.
- If desired, you can open the page in a text or HTML editor and move the slideshow HTML code block around.

Tip: To create additional slideshows, you'll just run the Standalone Slideshow Creator application again.

To edit a slideshow, you'll use a similar workflow:

- Re-launch the XFS v4 Standalone Slideshow Creator application, in the same manner as when you first created the slideshow.
- When prompted, specify the HTML page that holds the existing slideshow and the associated site root folder. In other words, just specify the same page and site root folder values as those you used when initially creating the slideshow.
- The external wizard will load in the slideshow in question, allowing you to make changes.
- When you're finished, you'll save changes and exit the external wizard.
- The Slideshow HTML and the XML configuration files will be updated based on any configuration changes you made

Accessing the Extension within Dreamweaver

This topic only applies to the Dreamweaver extension version of the product. If you're using the Standalone version, see this topic.

This section details how to access the extension within Dreamweaver, including:

- Inserting Slideshows
- Editing Existing Slideshows

Addressing Javascript Errors When Working with the Dreamweaver Extension

If the Dreamweaver extension behaves in an unpredictable manner or generates Javascript error messages when performing routine tasks, you likely have a corrupted Dreamweaver javascript cache. To reset the cache, do the following:

- Close Dreamweaver. This step is important!
- Browse to the Dreamweaver Configuration folder. Its location varies depending on the OS and the version of Dreamweaver:
  - Macintosh: /Users/{username}/Library/Application Support/Macromedia *or* Adobe/Dreamweaver {version}/Configuration/
    Important: Please note that this folder is NOT the /Library/Application Support/... folder; this is a separate folder.
  - Windows XP or 2000: C:\Documents and Settings\{username}\Application Data\Macromedia *or* Adobe\Dreamweaver {version}\Configuration\
  - Windows Vista or 7: C:\Users\{username}\AppData\Roaming\Macromedia *or* Adobe\Dreamweaver {version}\Configuration\
- The folder path will include *Macromedia* if you have Dreamweaver MX, MX 2004, or 8. Any later versions will include *Adobe* instead.
  The {version} value depends on your version of Dreamweaver; **for Dreamweaver CS3, it will be *Dreamweaver 9***.
- On Dreamweaver CS4, CS5 and higher, your path will likely include a language folder before the Configuration/ folder, like this: .../Dreamweaver {version}/*en_US*/Configuration... . The *en_US* value will depend on your Dreamweaver's localization.
- In the Configuration folder, locate the `MacFileCache-xxxxx.dat` file (Macintosh) or `WinFileCache-xxxxx.dat` (Windows). The 'xxxxx' value will be a random string of several characters.
- Remove this file

- Restart Dreamweaver. The cache file will be recreated, and the javascript errors should be disappear.

Note: By default, the settings folders mentioned above are hidden on Windows systems. To make it visible, you must do the following:

- Open any folder in the Windows Exploreer
- Select Tools > Folder Options.
- Select the View tab in the Folder Options dialog.
- Check the 'Show hidden files and folders' radio button.
- Click OK to apply the changes.
- If you have any trouble or the steps don't match your version of Windows, see this helpful tutorial: http://www.bleepingcomputer.com/tutorials/index.php?act=print&tut=62&client=printer .

Inserting Slideshows

Accessing Via the Insert Panel

This topic only applies to the Dreamweaver extension version of the product. If you're using the Standalone version, see this topic.

To access the Insert Wizard to create a new slideshow, there are two options. First, the **'XML Flash Slideshow v4 - Insert Slideshow'** button can be accessed in the **DWUser** section of the Insert Panel. The panel is accessed via Window > Insert. The specific appearance of the panel varies depending on your version of Dreamweaver and your configuration:

Dreamweaver CS4 and CS5

Dreamweaver CS3

Dreamweaver MX, MX 2004, and 8

Accessing Via the Insert Menu

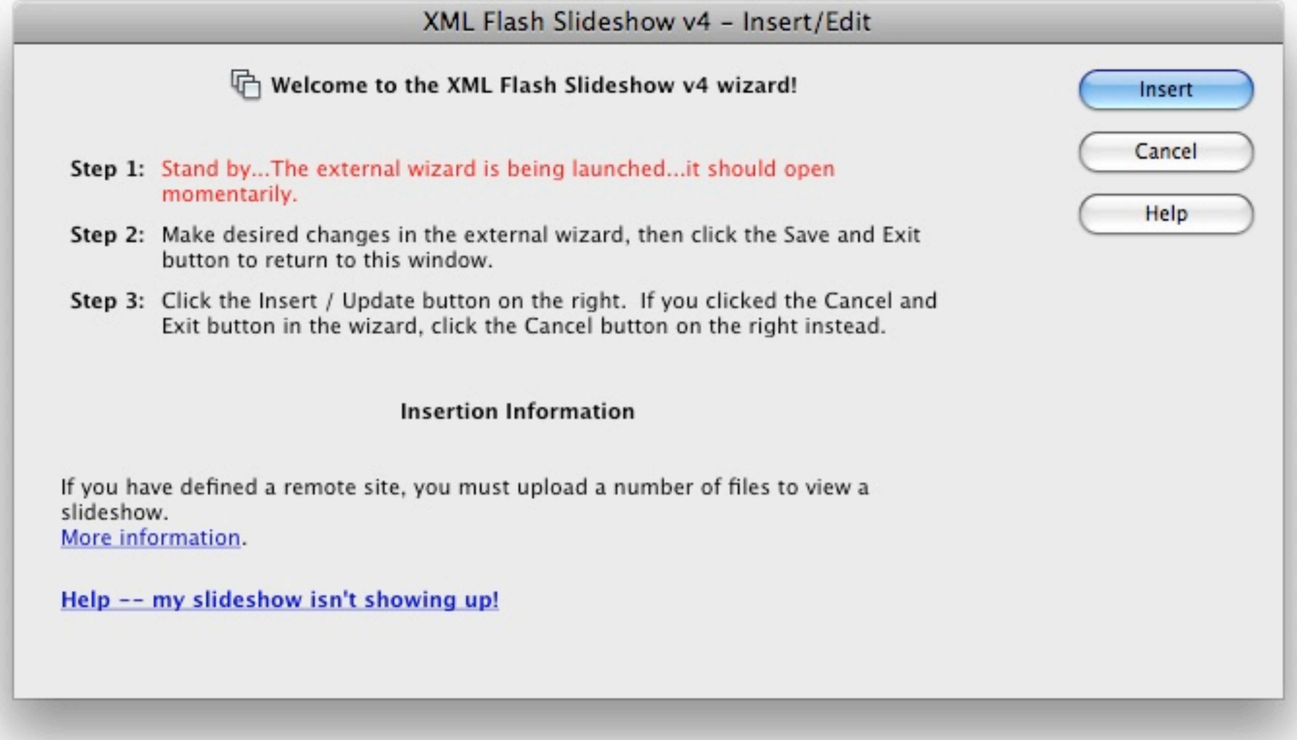
The Insert Slideshow wizard can also be accessed from the Insert Menu, which is the menu at the top of your screen next to File, Edit and View. It is accessed by selecting Insert > DWUser > XML Flash Slideshow v4.

Mac / Dreamweaver CS4/CS5 Screenshot

Windows Vista / Dreamweaver CS3 Screenshot

Once You Click The Insert Slideshow Button...

After you have clicked the Insert Slideshow button, the Dreamweaver extension dialog will momentarily appear:



Welcome to the XML Flash Slideshow v4 wizard!

Step 1: Stand by...The external wizard is being launched...it should open momentarily.

Step 2: Make desired changes in the external wizard, then click the Save and Exit button to return to this window.

Step 3: Click the Insert / Update button on the right. If you clicked the Cancel and Exit button in the wizard, click the Cancel button on the right instead.

Insertion Information

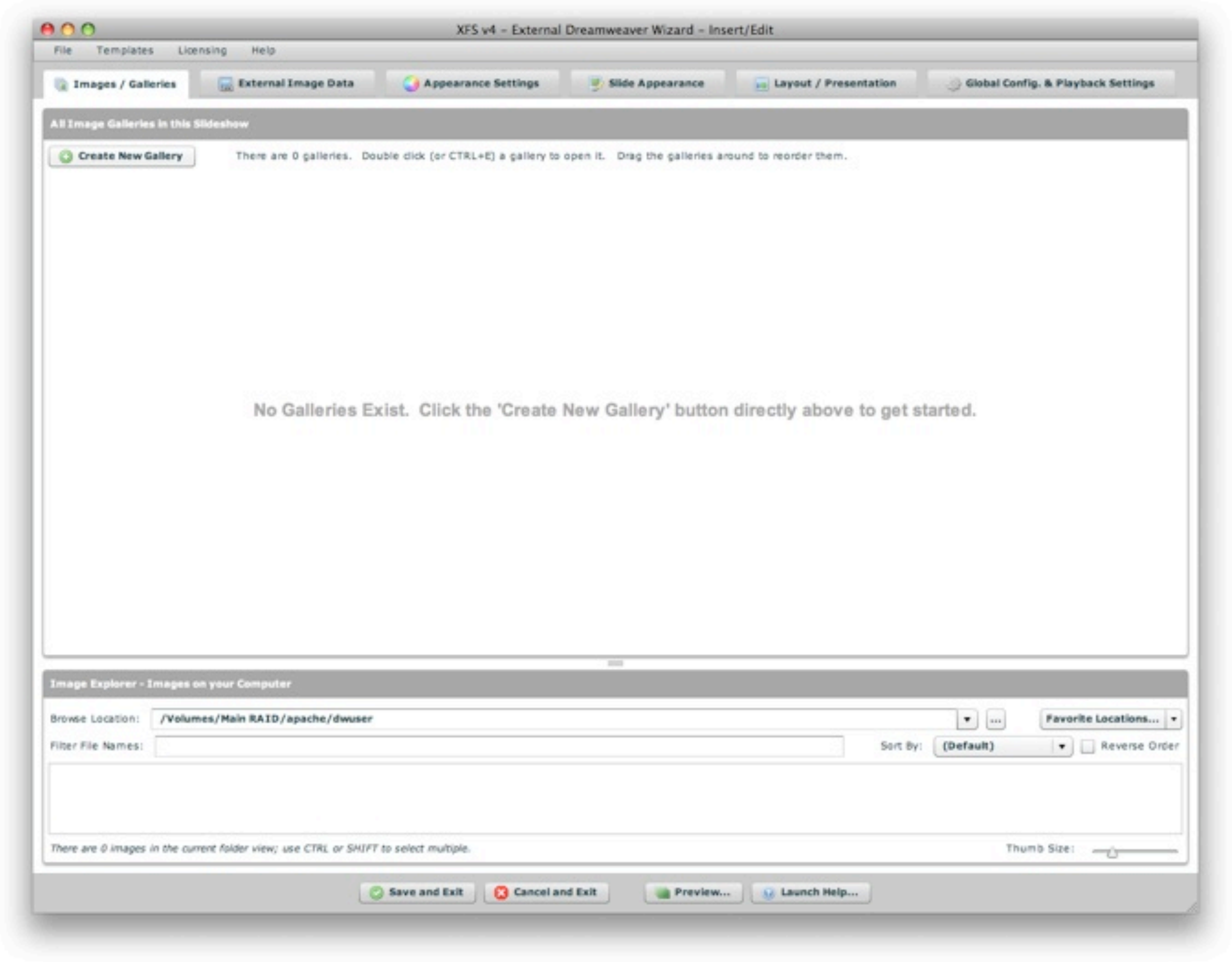
If you have defined a remote site, you must upload a number of files to view a slideshow.
More information.

Help -- my slideshow isn't showing up!

Note: If you installed the external wizard to somewhere other than the default location and the extension can't find it, you will be notified that it needs help locating it. If you receive this message, expand the following pod for details.

Manually Locating the External Wizard Application

After a moment, the external wizard window will open:



There are 0 galleries. Double click (or CTRL+E) a gallery to open it. Drag the galleries around to reorder them.

No Galleries Exist. Click the 'Create New Gallery' button directly above to get started.

Browse Location: /Volumes/Main RAID/apache/dwuser

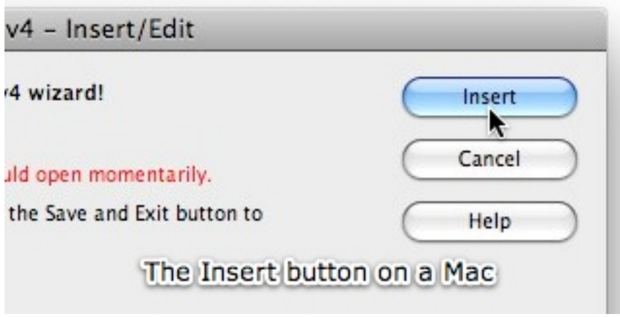
Filter File Names:

Sort By: (Default) □ Reverse Order

There are 0 images in the current folder view; use CTRL or SHIFT to select multiple.

This is where you will actually create the slideshow. When you are finished making edits in the external wizard, click the Save and Exit button:

You will return to the Dreamweaver extension window shown above; click the Insert button to actually insert the slideshow into your page:



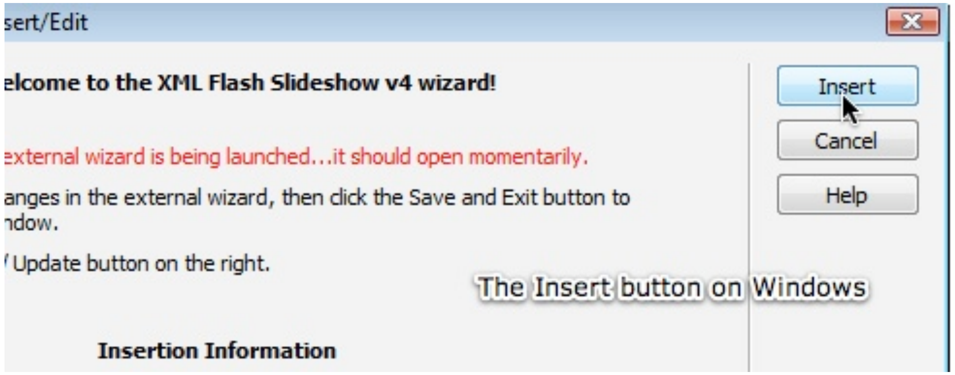
v4 – Insert/Edit

'4 wizard!

ıld open momentarily.

the Save and Exit button to

The Insert button on a Mac



sert/Edit

elcome to the XML Flash Slideshow v4 wizard!

external wizard is being launched...it should open momentarily.

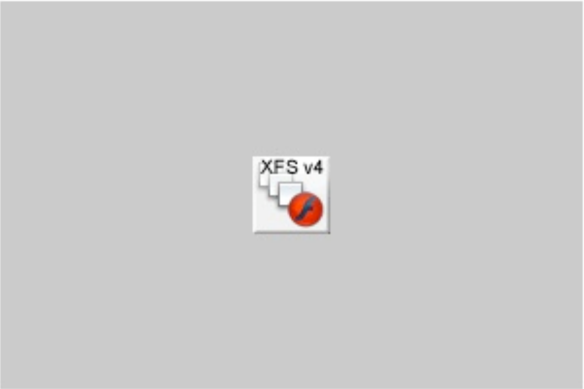
anges in the external wizard, then click the Save and Exit button to ndow.

' Update button on the right.

Insertion Information

The Insert button on Windows

Note that the process of actually **using the external wizard is detailed in subsequent topics**. Once you have inserted a slideshow, it will appear in Design View like this:



If you switch to code view, you'll be able to view the Slideshow HTML code which was inserted. At some point, you'll likely need to make edits; to see how to access the Edit Wizard, see the next topic: Editing Slideshows .

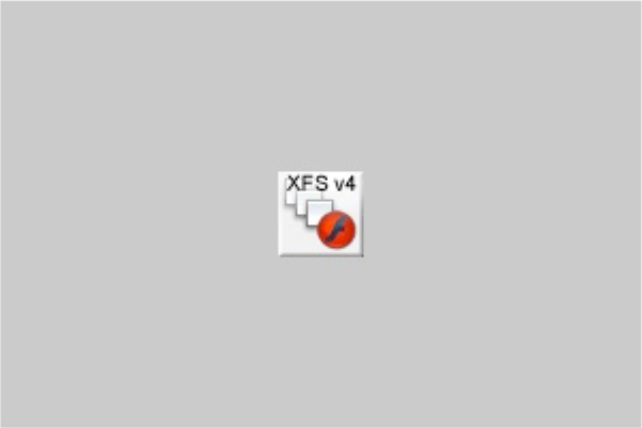
Creating Additional Slideshows

If you wish to create additional slideshows, simply run the Insert Slideshow wizard again (by clicking the Insert Slideshow button).

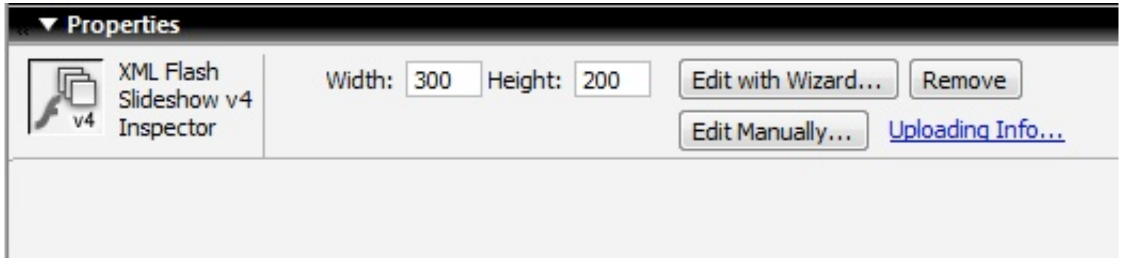
Editing Slideshows

Once you have created a slideshow, you will likely need to make changes at some point. To do this, you will use the Edit Wizard, which functions identically to the Insert Wizard except that it is pre-populated with all of the current slideshow's configuration settings.

After you have created a slideshow, you should see the following rendering in Design View:



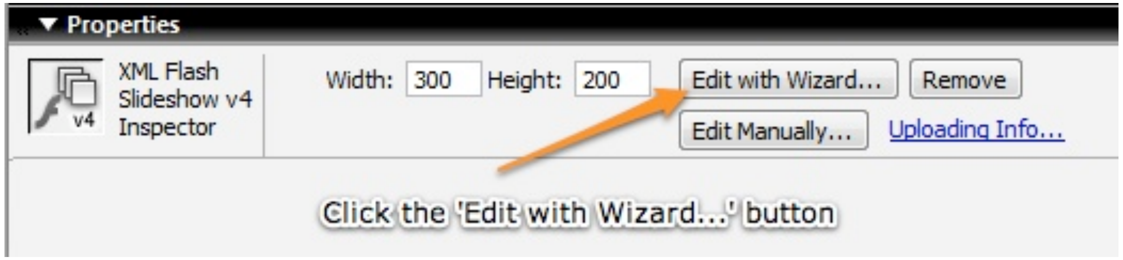
When you are working in Dreamweaver's Design View, this is the way your slideshows will appear. To actually view the slideshow, you must preview in the wizard or in a browser; this is explained in the Previewing Slideshows section of the documentation. To make changes, **click the rendering once to select the element**. In the Properties Panel (also known as the Property Inspector; accessed via Window > Properties), you should see the Slideshow Inspector:



Width: 300 Height: 200

Note - If you can't access the Slideshow Inspector, see the troubleshooting steps outlined below.

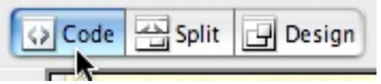
To launch the Edit Wizard, click the 'Edit with Wizard...' button in the Slideshow Inspector:

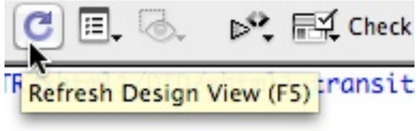


The 'Edit Manually...' button allows you to open the XML file for direct manual editing. The 'Remove' button will remove the slideshow HTML from your page. In a change from previous versions, the Remove button no longer deletes any XML files; if you wish to remove orphaned XML configuration files, you must do so manually. Clicking the 'Uploading Info' link will display information about which files must be uploaded to view a slideshow on a remote server in case you need a refresher.

Troubleshooting - Design View Rendering or Slideshow Inspector Doesn't Appear

Occasionally, the Design View rendering or Slideshow Inspector don't appear as they properly should. If you run into this problem, the following steps should correct it:

- Switch to Code View:
  

- Place your cursor somewhere within the slidehow HTML code block. This code block is denoted by `<!-- Start dwuser_XML_Flash_Slideshow_v4 -->` and `<!-- End dwuser_XML_Flash_Slideshow_v4 -->`.
- Enter a single space character, then use the backspace key to immediately remove it.
- Click the 'Refresh Design View (F5)' button at the top of the edit window:
  

- The Slideshow Inspector should now be appear in the Properties panel.

If for some reason the Slideshow Inspector still doesn't appear, uninstall and reinstall the extension. This should address the corruption in the installation. To uninstall and reinstall the extension, open the Extension Manager (you can do this from within Dreamweaver by selecting Help > Manage Extensions...). Next, close Dreamweaver. Then, select the XML Flash Slideshow Creator v4 item and click the delete button (the trash-can icon). The extension should uninstall and notify you of such. Then, click the Install button and browse to the `SSv4_DW_ext.mxp` file you originally installed and click OK; the extension will reinstall and you can restart Dreamweaver. As a general tip, this reinstallation process often addresses other instances of the extension malfunctioning; if something randomly starts malfunctioning, reinstalling is always a good idea.

Getting Started > **Accessing the Standalone Slideshow Creator Wizard**

Accessing the Standalone Slideshow Creator

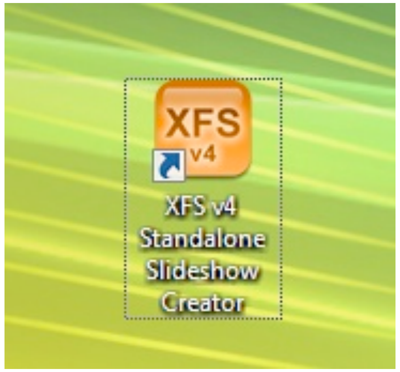
This topic only applies to the Standalone version of the product. If you're using the Dreamweaver Extension version, see this topic.

This section details how to access the standalone version of the product, including:

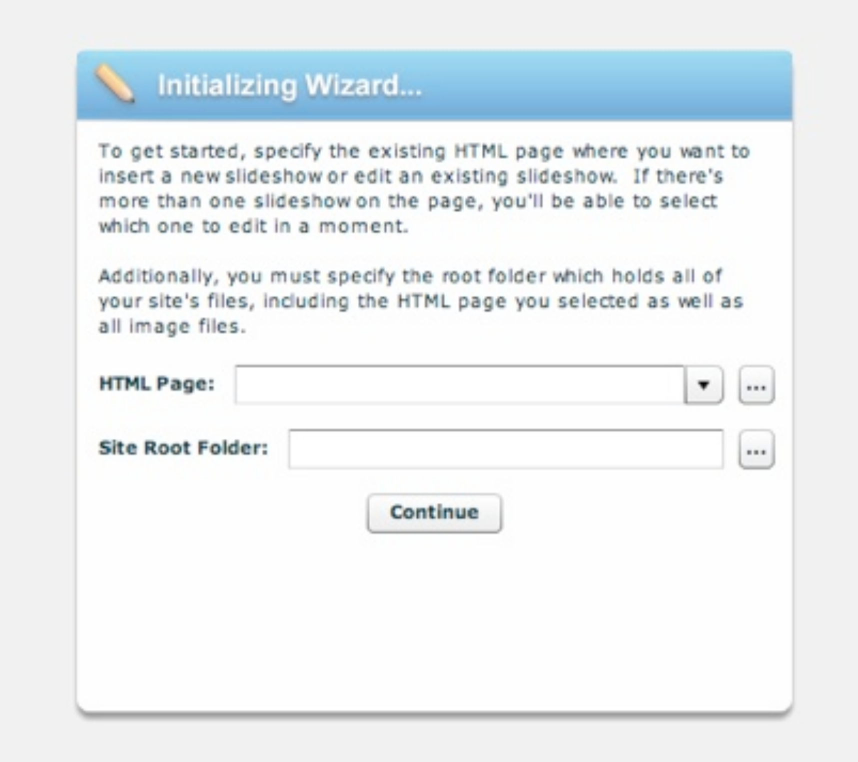
- Inserting Slideshows
- Editing Existing Slideshows

Inserting a Slideshow

To create a slideshow, double-click the *XFS v4 Standalone Wizard* icon on your desktop (or launch the program from the Programs / Applications menu):



This will launch the slideshow wizard. The first dialog will ask you to specify the URL of your page (where you want the slideshow to be inserted), and the root folder of the site associated with that page:



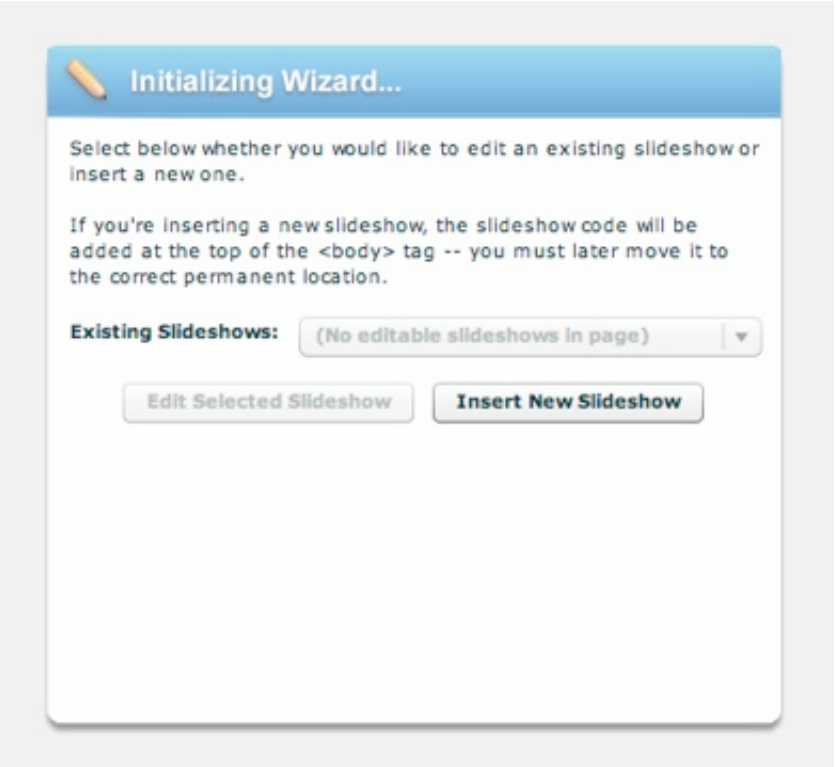
Initializing Wizard...

To get started, specify the existing HTML page where you want to insert a new slideshow or edit an existing slideshow. If there's more than one slideshow on the page, you'll be able to select which one to edit in a moment.

Additionally, you must specify the root folder which holds all of your site's files, including the HTML page you selected as well as all image files.

Note: Recent pages and their associated site root folders will automatically be saved; they can be accessed later by using the dropdown on the *HTML Page* field.

Next, you'll be asked to either create a new slideshow or edit an existing one. For new pages, you'll see the following:

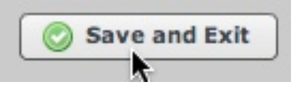


Select below whether you would like to edit an existing slideshow or insert a new one.

If you're inserting a new slideshow, the slideshow code will be added at the top of the <body> tag -- you must later move it to the correct permanent location.

Existing Slideshows: (No editable slideshows in page)

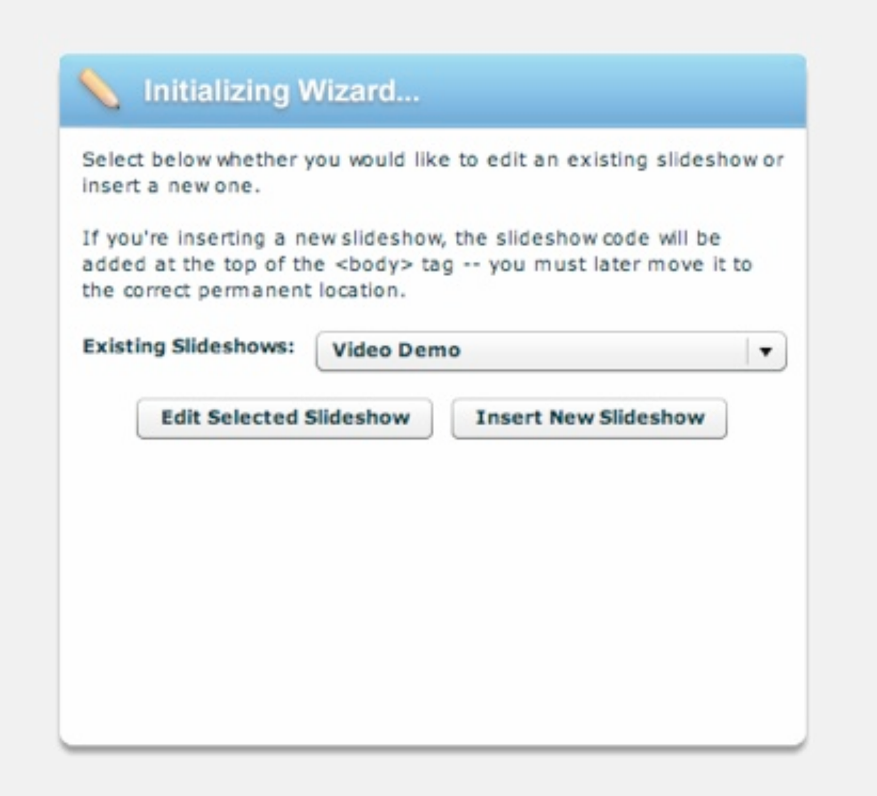
You'll then be ready to use the External Wizard to create your slideshow. When you are finished making edits in the external wizard, click the Save and Exit button:



This will close the external wizard and insert the necessary slideshow code to your page. If you open your HTML page in a text or HTML editor, you'll see that the slideshow code block was added at the top of the `<body>` tag. You can move the block of code around, positioning it as desired within your page.

Editing Existing Slideshows

Once you've added a slideshow to your page, you'll likely need to make edits at some point. To do this, re-launch the *XFS v4 Standalone Slideshow Creator* application as you did when initially creating the slideshow. In the initial dialog, select the HTML page that holds your slideshow. The associated site root will automatically be filled in. Clicking Continue, you'll now have the option to either insert a new slideshow or edit the existing slideshow(s):



Initializing Wizard...

Select below whether you would like to edit an existing slideshow or insert a new one.

If you're inserting a new slideshow, the slideshow code will be added at the top of the <body> tag -- you must later move it to the correct permanent location.

Existing Slideshows: Video Demo

The dropdown provides access to all slideshows in the page, listed by their name (a value specified when initially inserting the slideshow). The slideshows will be listed in the dropdown in the same order as they appear in the page. Selecting a slideshow and clicking the *Edit Selected Slideshow* button will load the slideshow into the wizard for editing. Again, when you're finished, click the Save and Exit button. The code in your page will automatically be updated.

Previewing Slideshows

This section details how you can preview slideshows before deploying them to a remote server. Proceed to the next subtopic to learn more.

Getting Started > Previewing Slideshows > **Previewing Locally - In the Wizard or a Browser**

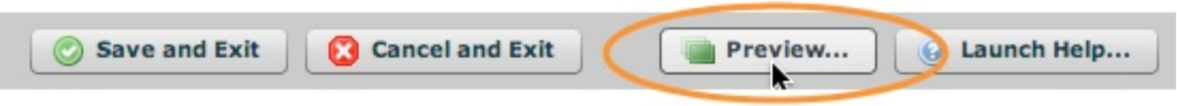
Previewing Slideshows Locally - In the Wizard or a Browser

When you are developing a slideshow, you'll likely want to preview your work as you make changes. There are two methods to accomplish this -- you can preview directly in the wizard using the Preview feature, or preview the file on your local machine in a browser.

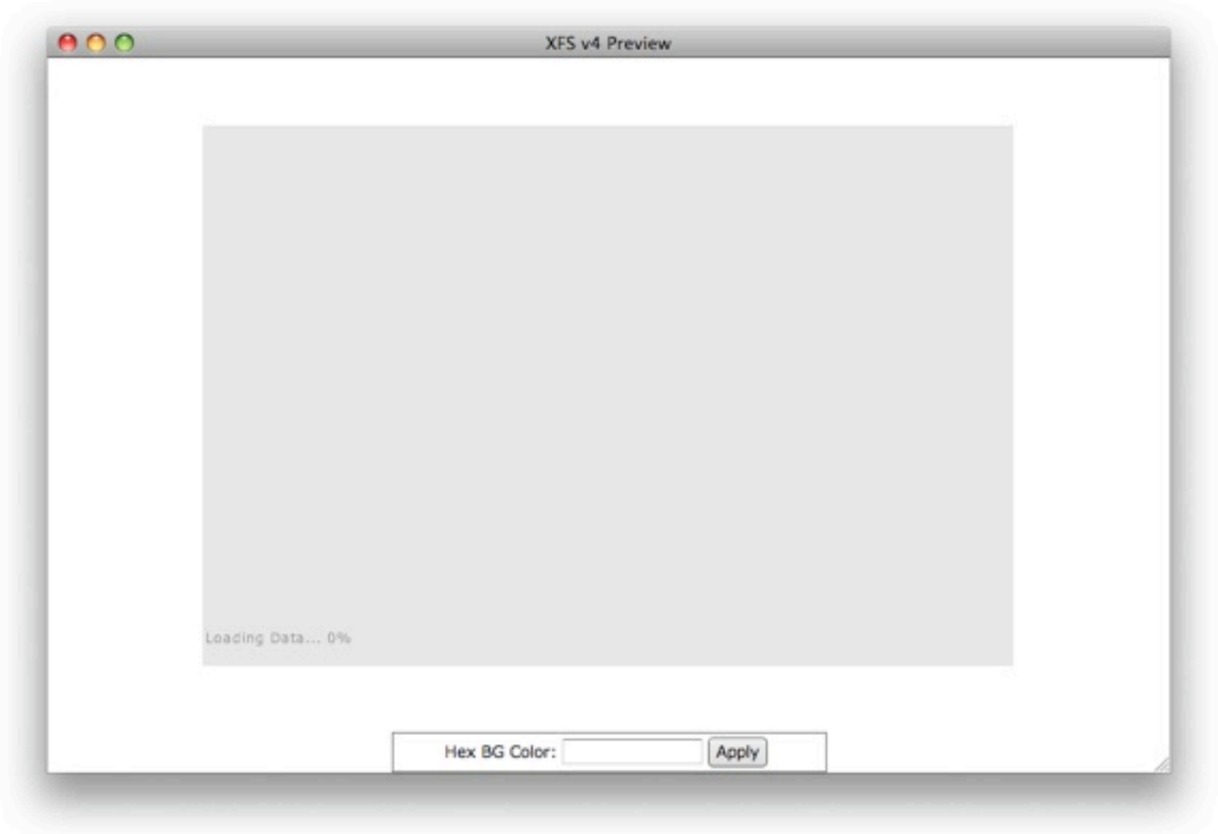
Previewing In the Wizard

One of the most important improvements in version 4 is the introduction of a new built-in Preview feature. This allows you as the designer and/or developer to preview your slideshow instantaneously as you make changes in the external insert/edit wizard without uploading any files or exiting the wizard. This improvement offers immense time savings.

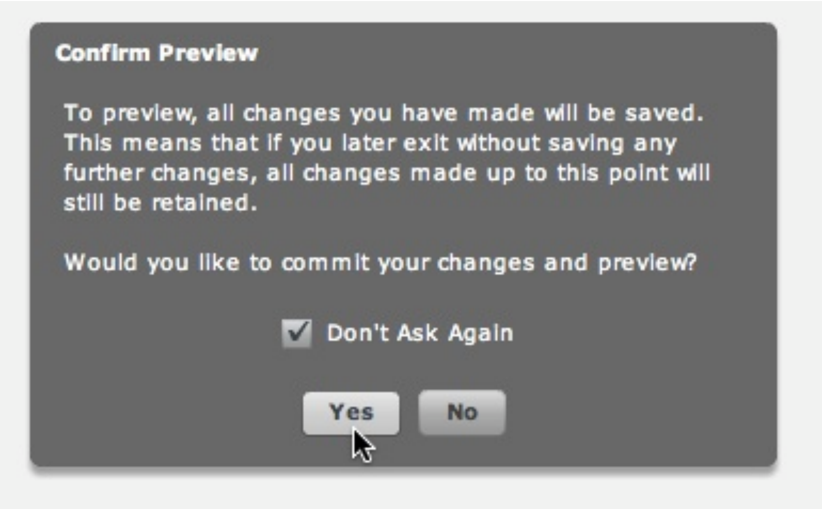
To preview a slideshow, you'll click the **Preview...** button at the bottom of the wizard:



This will launch the preview dialog, which is a separate application window; because it is a separate window, you can switch back to the main wizard and make more changes, then flip back without re-launching the preview. When you're ready to preview any new changes, click the Preview button again. This will re-launch the preview window.



The first time you launch the preview window, you will receive a message confirming that you want to commit your changes:



Confirm Preview

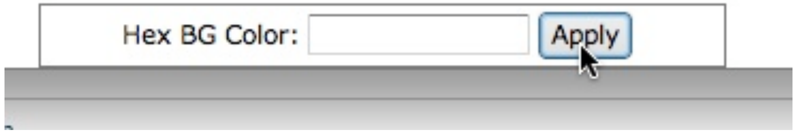
To preview, all changes you have made will be saved. This means that if you later exit without saving any further changes, all changes made up to this point will still be retained.

Would you like to commit your changes and preview?

☑ Don't Ask Again

This is because the changed data must be written to the XML configuration file in order to generate the preview. Thus, any changes which you have made which affect the external XML file will be applied. Therefore, cancelling out of the wizard after commiting data for a preview will still result in the changes being maintained. If you are making changes which you may want to later undo, you can always save the slideshow's settings as a template (see the Using Templates section of the manual). Then, if something goes wrong, you can revert back to the working version. If everything goes smoothly, you can remove the unneeded template. If you leave the 'Don't Ask Again' checkbox selected, you won't receive the message again when you click the Preview button for the duration of your wizard session.

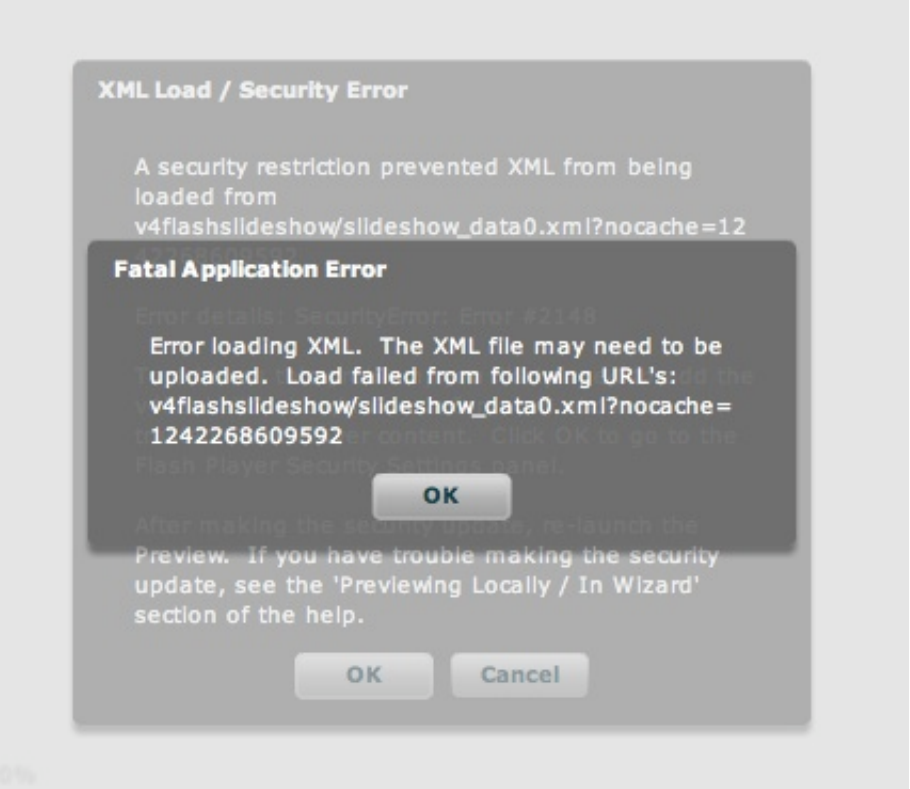
To **change the background color of the preview window**, enter a CSS color name or hex color window in the Background Color box at the bottom of the preview window and click Apply:



Keep in mind that this setting doesn't affect the slideshow or your HTML page in any way; it just allows you to customize the appearance of the preview window.

Setting Flash Player Security Settings

When you preview a slideshow a slideshow locally for the first time, you will likely receive two error messages, like this:



XML Load / Security Error

A security restriction prevented XML from being loaded from v4flashslideshow/slideshow_data0.xml?nocache=1242268609592

Error details: SecurityError: Error #2148

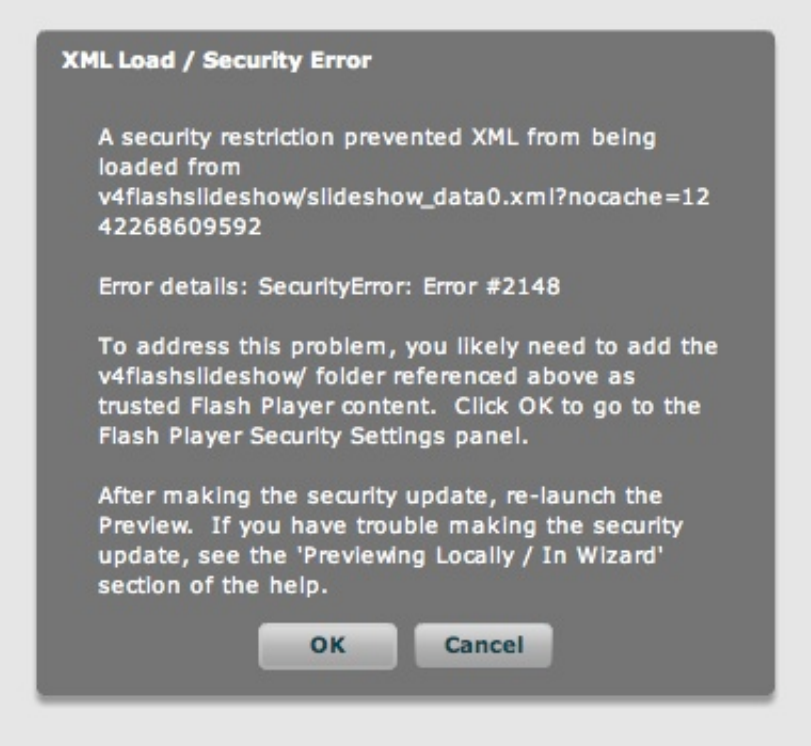
...dd the ...er content. Click OK to go to the Flash Player Security Settings panel.

After making the security update, re-launch the Preview. If you have trouble making the security update, see the 'Previewing Locally / In Wizard' section of the help.

Fatal Application Error

Error loading XML. The XML file may need to be uploaded. Load failed from following URL's: v4flashslideshow/slideshow_data0.xml?nocache=1242268609592

The first message says that the XML file failed to load. The second explains this error, stating that it is due to a Flash Player security restriction:



XML Load / Security Error

A security restriction prevented XML from being loaded from v4flashslideshow/slideshow_data0.xml?nocache=1242268609592

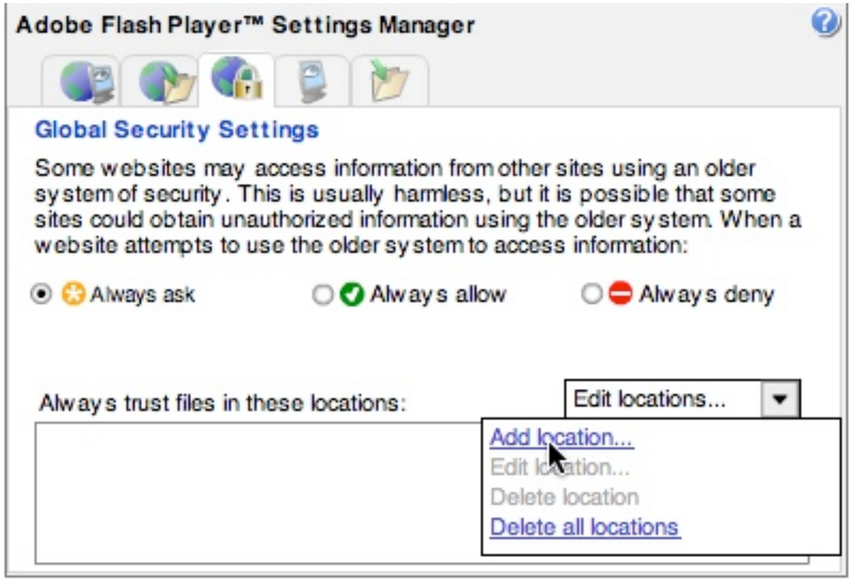
Error details: SecurityError: Error #2148

To address this problem, you likely need to add the v4flashslideshow/ folder referenced above as trusted Flash Player content. Click OK to go to the Flash Player Security Settings panel.

After making the security update, re-launch the Preview. If you have trouble making the security update, see the 'Previewing Locally / In Wizard' section of the help.

The problem is that when previewing locally (such as in the wizard, or in a browser from a file:/// based URL), the Flash Player by default won't let any SWF files load external data. This prevents the slideshow from being able to load the XML configuration file or any image files. Keep in mind that this **does not** affect any slideshows once they are viewed on a server, using an http://-based URL; it only affects local files.

The solution is to add the `v4flashslideshow/` supporting files folder as trusted content, so that the Flash Player knows it's OK to let it load external data. You can go directly to the Flash Player security settings panel by clicking the OK button in the message above; you will be automatically redirected to: http://www.macromedia.com/support/documentation/en/flashplayer/help/settings_manager04.html . In the settings manager, you will need to add the v4flashslideshow/ folder as a trusted location. To do this, click the **Add Location** link on the dropdown menu:



Adobe Flash Player™ Settings Manager

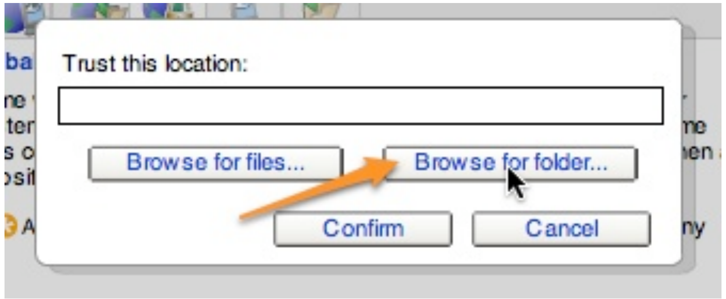
Global Security Settings

Some websites may access information from other sites using an older system of security. This is usually harmless, but it is possible that some sites could obtain unauthorized information using the older system. When a website attempts to use the older system to access information:

● Always ask ○ Always allow ○ Always deny

Always trust files in these locations:

In the browse dialog, click **Browse for Folder...**:



Trust this location:

Browse to the `v4flashslideshow/` supporting files folder in your site (referenced in the error message above) and click OK. The folder should appear in the list of trusted files.

Tip: On Macintosh systems, you may need to manually edit the path to include `/Volumes/Macintosh HD/` or similar (based on your actual path). What's important is that the URL **exactly** matches that which is displayed in the browser when previewing.   Otherwise, the Flash Player won't know to trust the content, and the preview will fail.

Tip: On non-english systems where the path includes accented characters, you may need to manually edit the path, replacing the accented character(s) with their non-accented counterparts.

Tip: If you perform the above steps and you still receive the error when previewing, check the tip above. If that's not the issue, try re-loading the security settings panel. If the folder you just added is NOT there, and won't persist when you add it again, this indicates that your Flash Player settings folder has become corrupted. The solution is the remove the current Flash Player settings folder, which forces a new clean one to be created:

- On Windows 2000 / XP, remove this folder: C:\Documents and Settings\{username}\Application Data\Macromedia\Flash Player\
- On Windows Vista or 7, remove this folder: C:\Users\{username}\AppData\Roaming\Macromedia\Flash Player\
- On Macintosh systems, remove this folder: /Users/{username}/Library/Preferences/Macromedia/Flash Player/
  (Note that this is NOT the /Library/ folder, but /Users/{username}/Library/...).

Previewing Locally in a Browser

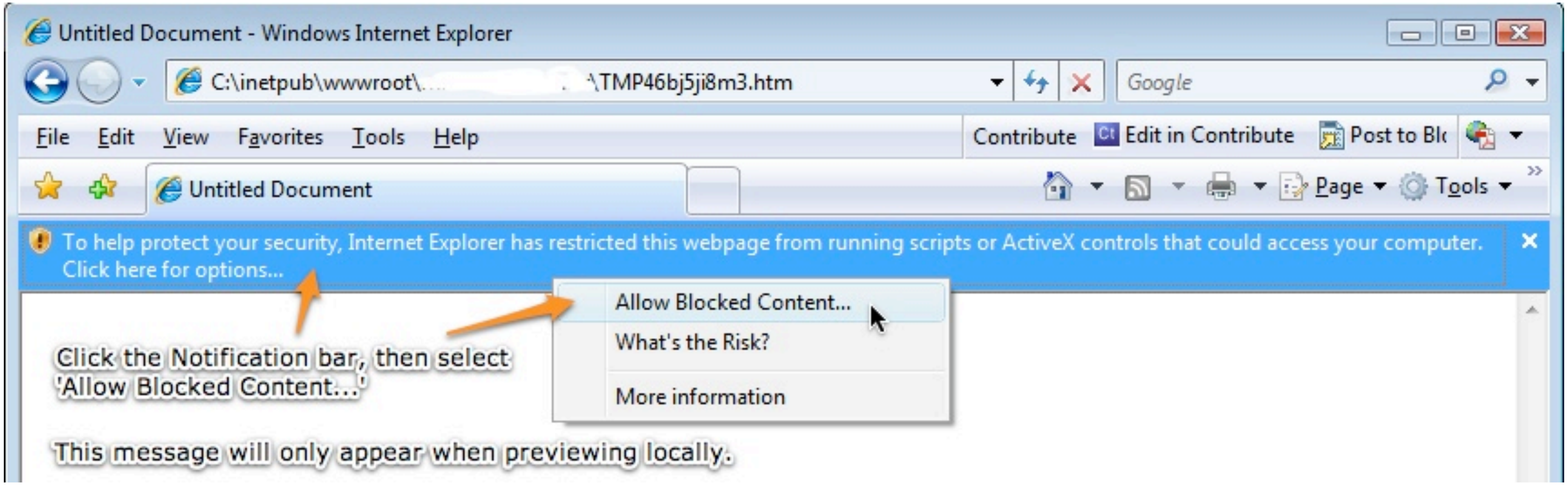
If you wish to preview the slideshow in context, within your page, you need to save your changes in the external wizard and insert or update the slideshow. This will cause the latest data to be committed to file and the latest slideshow HTML to be added to your HTML page. Once you have done this, you are ready to preview the slideshow.

To ensure that the preview displays properly in the browser when using Dreamweaver's Preview In Browser feature, it is important to **disable** the **'Preview In Browser' > 'Preview Using Temporary File'** setting in the **Dreamweaver Preferences dialog**. Failing to do this can cause the slideshow to appear as a gray square with a 'broken image' icon in the corner. To make this setting change, go to Edit > Preferences... (Windows) or Dreamweaver > Preferences... (Mac), select the Preview in Browser section, then disable the 'Preview Using Temporary File' option:

Screenshots for Mac

Screenshots for Windows

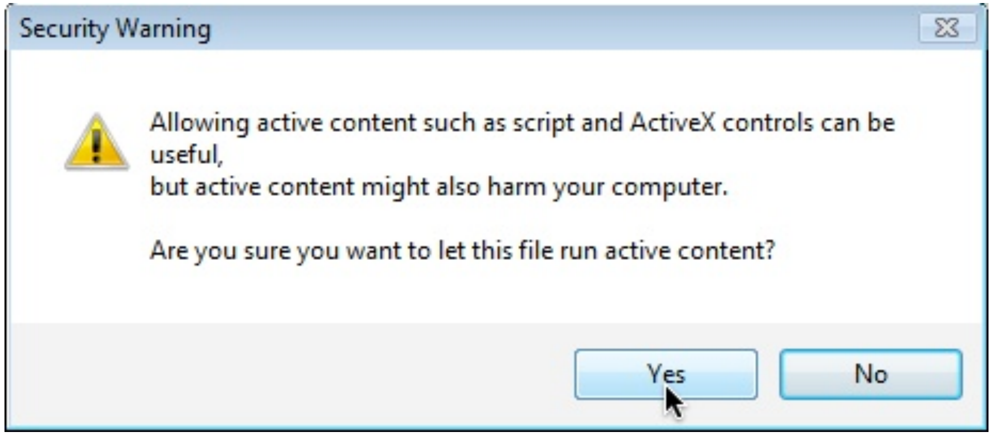
You can now use the Preview in Browser feature and the slideshow should appear properly. If you receive **error messages about Flash Player security**, see the section above on Flash Player Security Settings. If you are using Internet Explorer on Windows, you may receive a security warning about running scripts or ActiveX content. As with the Flash Player security restrictions, these **do not** appear once you view the slideshow from an http://-based URL after uploading; they only affect local previews from file://-based URLs. To make IE allow the the slideshow to appear, click the notification bar, select 'Allow Blocked Content...', then click Yes:



To help protect your security, Internet Explorer has restricted this webpage from running scripts or ActiveX controls that could access your computer. Click here for options...

Click the Notification bar, then select 'Allow Blocked Content...'

This message will only appear when previewing locally.



Allowing active content such as script and ActiveX controls can be useful,
but active content might also harm your computer.

Are you sure you want to let this file run active content?

Previewing Remotely

If you wish to preview a slideshow on the remote server, follow the same steps as you would when deploying the slideshow. This process is detailed in the Deploying Slideshows topic.

Deploying Slideshows

Once you've created your slideshow and made all desired customizations, the time will come when you're ready to put it live on the web. The process of uploading slideshows to a remote server is relatively simple, but you must follow the steps outlined below for the slideshows to appear properly.

First, it is important to note that there are **no** server-side requirements. You can deploy slideshows to any type of webserver.

Slideshow File Structure

When you insert a slideshow, several external files are referenced directly or indirectly by the slideshow HTML. Because paths to these files are constructed based on their location relative to the HTML page, it is important to have them in place *before* you insert the slideshow, and not modify the paths after you create the slideshow. The relevant files include:

- The files in the `v4flashslideshow/` supporting files folder. The location of this folder is specified when you first create a slideshow for the site.
- The image files. You can place them anywhere within your site; they do not have to be within a specific folder. However, it is required that they are placed within your Dreamweaver site's web root. This means that they must be accessible from the web when you upload them. For example, if your site root contains an `htdocs/` folder and all of your actual webpages go within this folder, your image files must be within this folder as well. While not specifically required, best practices dicatate that you place all images within a dedicated folder, such as an `images/` folder.
- Any audio or video files used in the slideshow. The same location restrictions apply as for the image files.

Uploading Files

It is important to understand that all of the dependent files are loaded externally at runtime; they are not embedded in the slideshow SWF file. This ensures the best user experience, reduces loading time and provides maximum flexibility for you, the developer. Because of this, all of the following dependent files must be uploaded in addition to your HTML page for a slideshow to appear properly when deployed:

- All files in the `v4flashslideshow/` supporting files folder. The location of this folder is specified when you first create a slideshow for the site. Failing to upload these files will result in a 'script not found' message, a blank space where the slideshow belongs, or an XML loading error message.
- All image files, including any separate thumbnail images. Failing to upload the image files can result in '**Loading Error - IO**' messages appearing in place of the correct images.

- Any audio or video files used in the slideshow. Failing to upload these files will result in loading error messages.
- If using an external skin file, this skin file must be uploaded as well. Failing to upload the specified external skin file will will result in error messages.

Troubleshooting Common Issues

If **you receive a javascript alert message about the `v4flashslideshow/slideshow.js` file** not being accessible, make sure that you uploaded all files in the v4flashslideshow/ supporting files directory. If the problem persists, make sure that this line in your slideshow HTML properly points to the slideshow.js file (e.g. its path is correct):

```
<!-- Do not remove the line below!!! It is required for the DWUser XML Flash Slideshow v4. -->
<script type="text/javascript" src="v4flashslideshow/slideshow.js"></script>
```

If a **blank space appears when you try to view your slideshow** on the remote site, right-click on the blank space. Usually, you will see a 'Movie Loading...' message. If this message remains for more than just a few seconds, the issue is that the SWF file (in the `v4flashslideshow/` folder) was not properly uploaded. Try uploading it again. If you are using an external FTP program instead of Dreamweaver, make sure that you are uploading the SWF files in Binary mode, not ASCII mode.

If, in place of one or more images, **you see a 'Loading Error - IO' message**, this indicates that the images were unable to be loaded. Usually, this is because the image files must be uploaded. If the images still don't appear, there may be a path problem. You can check the image paths by doing this: 1 - Open the XML file in your browser. 2 - Copy and paste an image path (including the gallery's `fullPathPrefix` value if one exists) into your browser. It's important to construct the URL as if you were starting at the HTML page which holds the slideshow (NOT the XML file). 3 - Load this URL. If the image doesn't display properly (e.g. returns a 404 not found error), then double-check that your images have been uploaded and that the paths are correct. If the image displays properly, then there is some other problem.

If you are 100% sure that you have uploaded the images and the paths are correct, yet **you still see a 'Loading Error - IO' message** OR **you see a solid red square** in place of where the image(s) should be, this usually indicates that a permissions error is occuring behind the scenes, resulting in a 403 forbidden error when the images are requested. The 'solid red square' issue can also be caused by images which have been uploaded using an external FTP client using ASCII mode; Binary mode must be used to upload images. To address the permissions error, make sure that the file is accessible.

If you have an **error like the previous two, but it only seems to affect a single browser**, first try clearing your temporary internet cache. If the problem is still there, see if your webhost has a 'hotlink protection' setting. If so, try disabling it for the images in the affected slideshow. This setting can cause the server to misinterpret the image requests from the slideshow SWF based on the way certain browsers handle the HTTP request headers; the result is blocked images.

Loading any assets (XML, images, audio, video, etc) from a different domain (or a different server) can cause problems as well; see the Flash Player Security Restrictions section below.

General Tip: If you make changes but they don't seem to be rendered when you view the slideshow in your browser, you likely need to clear your temporary internet files. Old cached versions of various files can prevent new updates from showing.

Flash Player Security Restrictions and Cross-domain Loading Errors

If you are loading image files (or video or audio) or external image data from a server at a different domain, you will be affected by Flash Player Security Restrictions. In a nutshell: the Flash Player will not allow content to be loaded from a different domain than the residing SWF file sits on, unless the owner of that different domain specifically authorizes this loading.

First, it's important to explain what qualifies as a 'diffent domain'. Any difference in either a subdomain or root domain qualifies a URL as being an entirely different domain. The most common example might surprise you:

You have a slideshow on http://**www.example.org**/slideshow.html. You upload some images to your site, and add them to the slidehow with a URL 'http://**example.org**/images/image1.jpg'. When you view the slideshow, the image fails to load and you receive a security restriction error message. "Huh?" you say, "I'm loading from the same site!" The issue is that the Flash Player views **example.org** and **www.example.org** as two separate domains because their subdomains aren't identical.

More intuitive examples include a slideshow at http://www.example.org/ which loads data from http://api.flickr.com/bla . Because this is a different root domain, Flash Player security restrictions go into effect.

When you attempt to load assets (XML data, images, audio, or video files) from a different domain, the Flash Player checks to see if a crossdomain policy file has been placed on the remote server. This is a file which is named **crossdomain.xml** and sits at the root of the domain's server (e.g. http://data.example.org/crossdomain.xml or http://www.dataserver.com/crossdomain.xml). This file must either allow all domains (via the use of a wildcard) or specifically allow the domain on which your slideshow resides, or the loading will not be permitted.

To prevent loading errors (as evidenced by **'Image Failed - Security'** messages and other loading errors, such as XML loading errors), you must either load from a server which specifically permits crossdomain loading (if you don't have control over the server), or place a crossdomain.xml file on the server. Here is a simple example file, which permits loading from across all subdomains on this domain:

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM
"http://www.adobe.com/xml/dtds/cross-domain-policy.dtd">

<cross-domain-policy>
<allow-access-from domain="example.com"/>
<allow-access-from domain="*.example.com"/>
</cross-domain-policy>
```

Two live example files can be viewed at: http://search.yahooapis.com/crossdomain.xml (an open policy, with no restrictions) and http://www.amazon.com/crossdomain.xml (a more restricted policy). You can learn more about creating crossdomain.xml files (crossdomain security policy files) at Adobe's site: http://www.adobe.com/devnet/articles/crossdomain_policy_file_spec.html

Advanced users who are comfortable with server-side programming can also circumvent these restrictions through the use of a server-side proxy on the same server as hosts the slideshow. For an example, see this blog post: http://blog.nelsonsilva.eu/2008/02/27/apache-proxy-for-crossdomain-problems/ .

Creating Custom Alternate Content

If a user doesn't meet the necessary requirements to view a slideshow (as detailed below; usually due to Javascript being disabled or an insufficient Flash Player version), the user will see the alternate HTML content. This is specified through the following code in the slideshow HTML code block:

```
<div class="dwuser_xfs_v4_holder" style="width: 600px; height: 400px;"> <strong><a
href="http://www.adobe.com/go/getflashplayer/">You need to upgrade your Flash Player and enable Javascript to
view this content &raquo;</a></strong> </div>
```

The default alternate content is a simple message which states "You need to upgrade your Flash Player and enable Javascript to view this content", and links to the Flash Player download page on Adobe.com. However, you can replace this default value with any desired HTML content, including images. Or, you could provide more content-specific information such as "Please enable javascript and upgrade your Flash Player to view the slideshow".

For more detailed information and examples, see this topic in the Developer Reference.

Deploying to a Blog or CMS

A special technique must be used when deploying slideshows to pages over which you don't have full control, such as a blog or CMS. To learn more about this situation, see Deploying to a Blog or CMS.

User Requirements

Many developers ask -- "What requirements are there for users to view slideshows I create?". The answer depends on what slideshow SWF type you've selected - Basic or Pro (available only in the Professional version of the software). If using the Basic version, the following requirements are in effect:

- Javascript must be enabled in the user's browser
- The Flash Player **version 9** or higher must be installed in the user's browser

If using the Pro version, the following requirements are in effect:

- Javascript must be enabled in the user's browser
- The Flash Player **version 10** or higher must be installed in the user's browser

In either case, if the user doesn't have the required Flash Player version but does have version 6.0.65 or higher installed, the Flash Player Express Install feature is automatically invoked. This will automatically upgrade the user's Flash Player and return them to your page.   If they don't have this version of the Flash Player (a miniscule fraction of users) or if they have Javascript disabled, the alternate HTML content will be displayed. You can customize this alternate content as described in the Creating Custom Alternate Content section above.

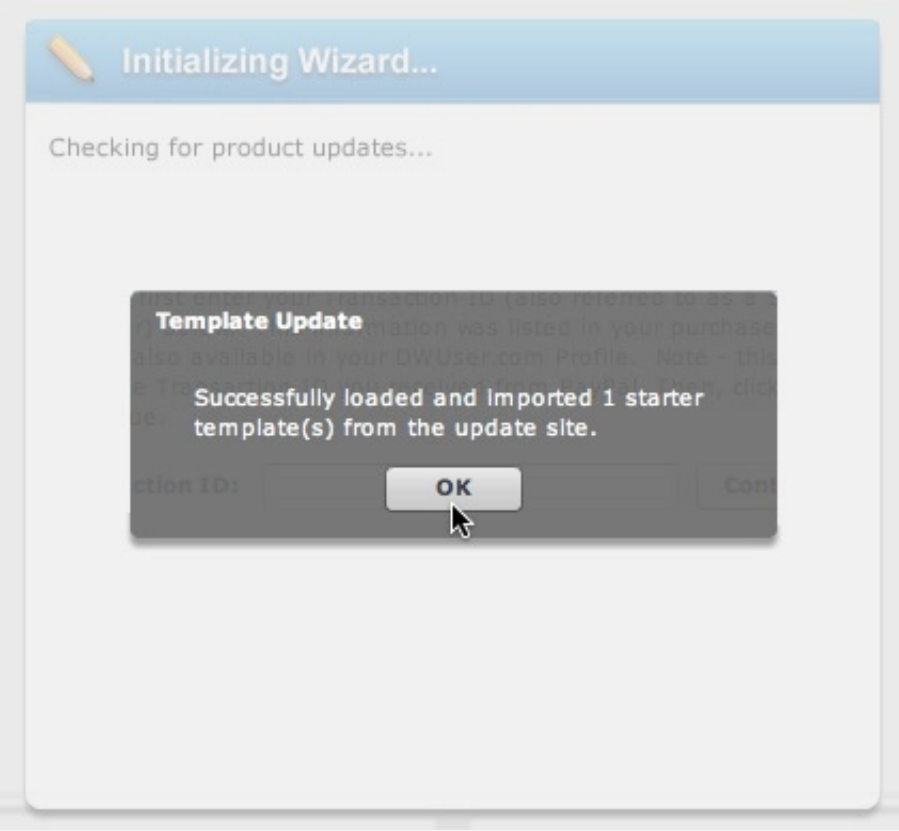
Using the External Wizard to Create and Edit Slideshows

The external wizard is where you will create and edit slideshows. (The process of launching the external wizard is detailed in the Accessing the Extension section.) This section of the manual covers all aspects of using the wizard in detail. The first few topics cover the initialization options; the section-by-section usage guide covers each of the different areas of the wizard.

Product Updates

When the wizard is launched for the first time, and periodically thereafter, the wizard will check to see if there are any important program updates available. Additionally, on the first launch, all starter templates will be downloaded into the program.

Tip: If your firewall software prompts you to give the program access to the internet, be sure to allow the communication.



Initializing Wizard...

Checking for product updates...

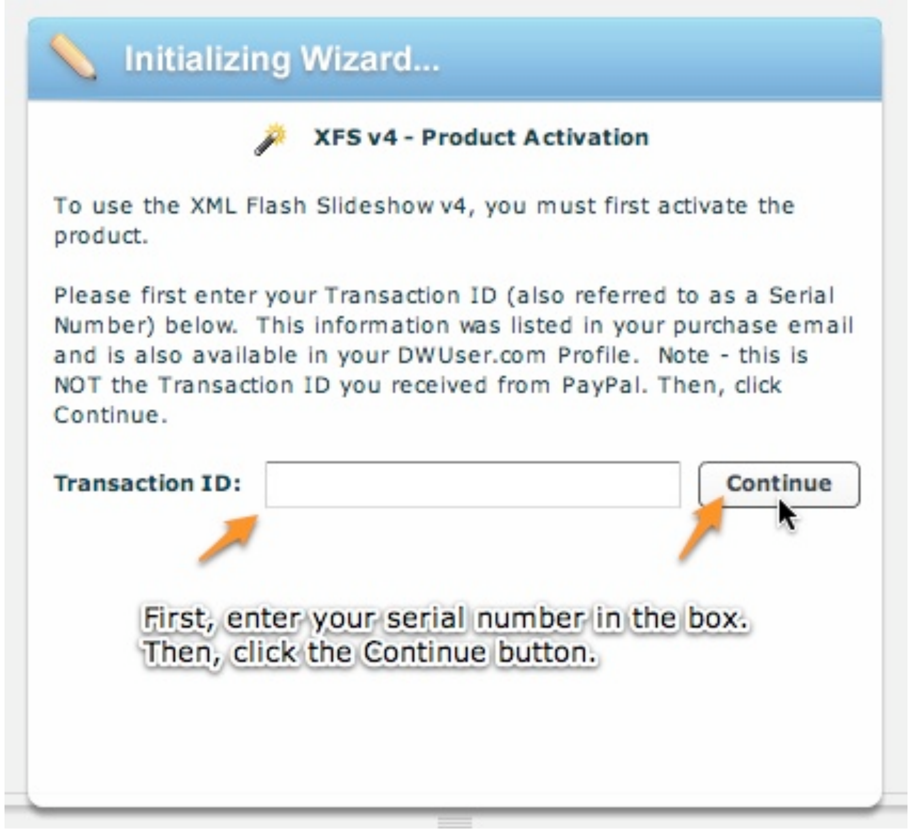
Template Update

Successfully loaded and imported 1 starter template(s) from the update site.

If there are any updates available, you will be notified and a link to the update page will be shown. If there are any starter templates available, they will be downloaded and imported and you will be notified. This process will only take a few seconds.

Product Activation

The first time you launch the external wizard, you must activate the product. The activation dialog looks like this:



XFS v4 - Product Activation

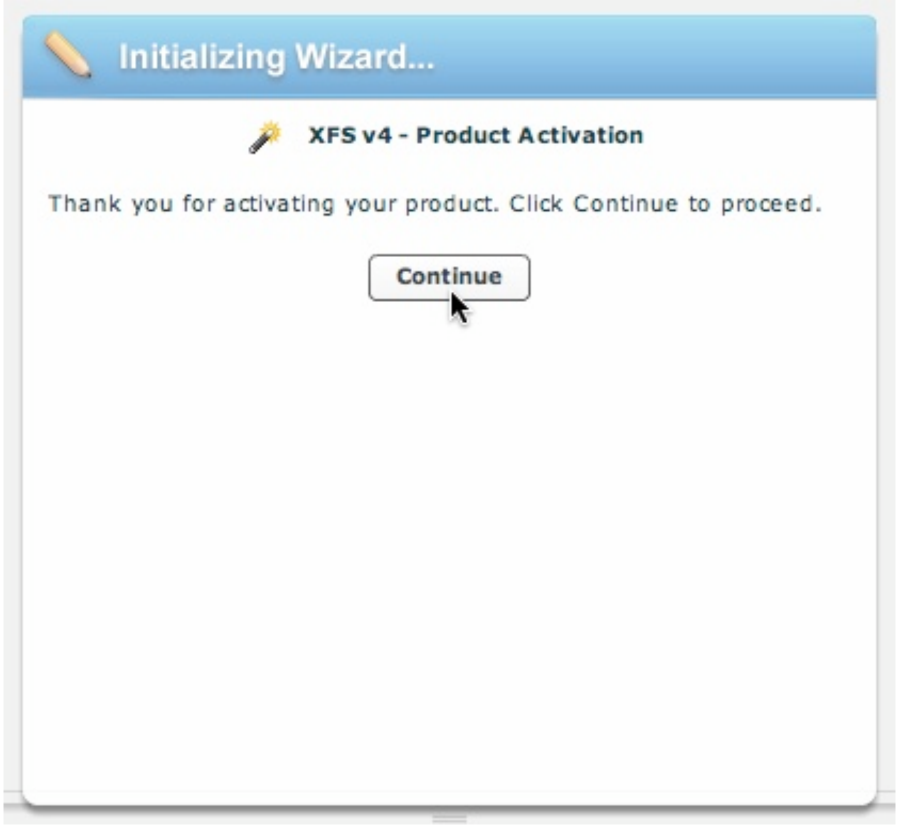
To use the XML Flash Slideshow v4, you must first activate the product.

Please first enter your Transaction ID (also referred to as a Serial Number) below. This information was listed in your purchase email and is also available in your DWUser.com Profile. Note - this is NOT the Transaction ID you received from PayPal. Then, click Continue.

Transaction ID:

First, enter your serial number in the box.
Then, click the Continue button.

First, enter your serial number in the box. This number will be listed in your DWUser.com profile, and was also sent to you via email when you purchased the product. (Note - This serial number is also sometimes referred to as a Transaction ID; these numbers are the same thing. This is NOT the transaction ID you recieved from PayPal if you purchased via PayPal.) The activated version of the product (Basic or Pro) will depend on the license type you purchased. After entering the serial number, click the Continue button. The product will activate over the internet and notify you when it completes:



Initializing Wizard...

XFS v4 - Product Activation

Thank you for activating your product. Click Continue to proceed.

Click **Continue** to proceed with the initialization.

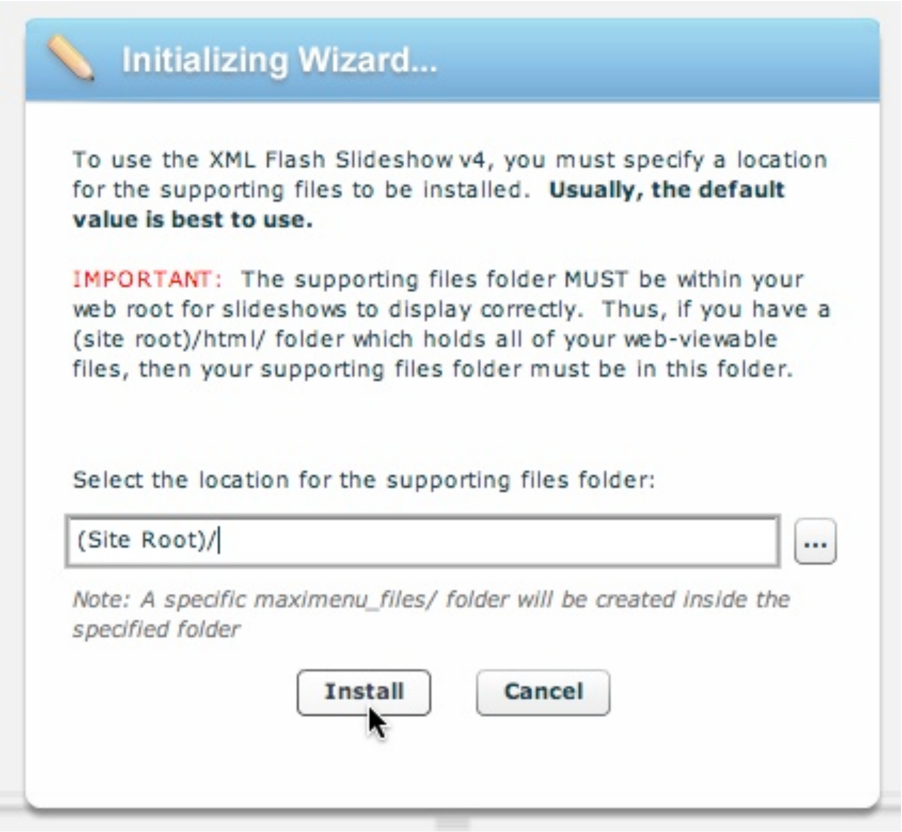
Important Note: For the program to function properly, it must be granted full internet access. In addition to activating, this allows it to download the starter templates (none actually ship with the product) and detect when updates are available. If you have trouble activating, make sure that your firewall is configured to allow the program to access the internet.

Supporting File Installation

For slideshows to be displayed on your site, a folder of supporting files must be copied into your site. The files can be placed anywhere in your site that is directly above the HTML page in question, up to the site root. A specific `v4flashslideshow/` folder will be created in the directory you choose, and this folder will hold all of the supporting files.

The Dreamweaver site's root root folder (the "site root") is the default location for the supporting files folder. (This results in the files being placed at, for example, `(site root)/v4flashslideshow/slideshow.js`). **IMPORTANT:** This default location is appropriate for most setups. However, if your site root is *not* the same as your web root, you should install the files to your **web root** instead of your site root. For example, some webservers require that you place all web-accessible files in the `htdocs/` folder. In this example, the file at `(site root)/htdocs/index.html` would map to `http://www.mydomain.org/index.html` . In this type of setup, you must select the `(site root)/htdocs/` folder as the installation location. If you don't place the supporting files in the web root (the `htdocs/` folder in this example), they will be stored on the server, but won't be accessible from the web.

The Supporting Files Installation dialog appears as follows:


Initializing Wizard...

To use the XML Flash Slideshow v4, you must specify a location for the supporting files to be installed. **Usually, the default value is best to use.**

IMPORTANT: The supporting files folder MUST be within your web root for slideshows to display correctly. Thus, if you have a (site root)/html/ folder which holds all of your web-viewable files, then your supporting files folder must be in this folder.

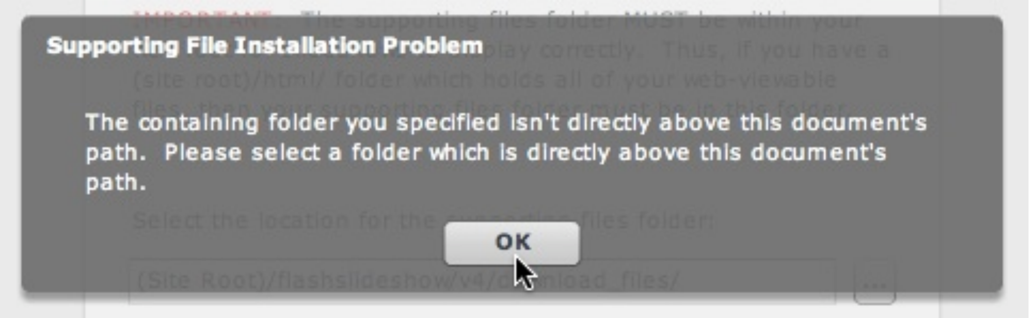
Select the location for the supporting files folder:

(Site Root)/

Note: A specific maximenu_files/ folder will be created inside the specified folder

Choose an appropriate location, then click **Install** to continue.

Note: If you choose a location which is not directly above the current page, or is not in the site's root, an error message will be displayed and you will be required to select a valid location:



Supporting File Installation Problem

The containing folder you specified isn't directly above this document's path. Please select a folder which is directly above this document's path.

Note: If, when inserting your first slideshow, you install the supporting files to a location which is not directly above another page in your site, you can still add a slideshow to that page. When you insert a slideshow into that second page, you will again be prompted to install the supporting files. Choose a location which *is* directly above that second page, and an additional copy of the supporting files will be installed to your site. When uploading files, just make sure that you upload both `v4flashslideshow/` folders.

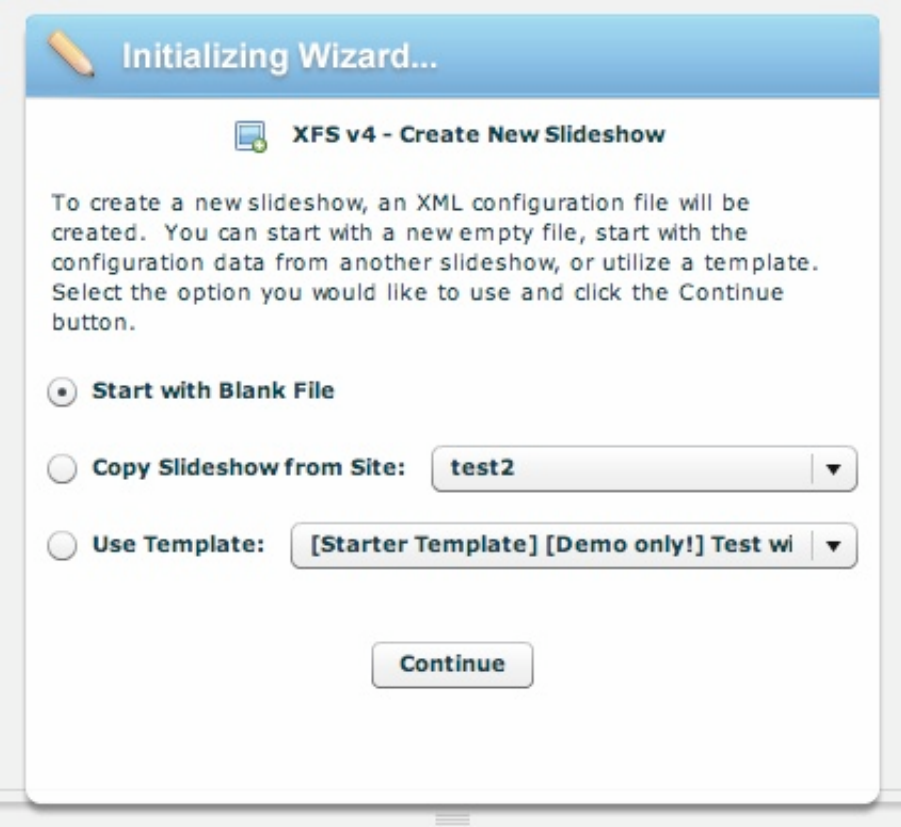
Tip: If you mis-install the supporting files, or need to change their installation location at some later time, you can do so by simply moving the folder. However, you must also update all of the script references in the slideshow HTML code blocks to point to the new location:

```
<script src="....v4flashslideshow/slideshow.js"></script>
```

If you don't update these script references, you will receive an error message about the script file being inaccessible when attempting to view the slideshows.

Creating a New Slideshow Configuration File

If you are running the Insert Wizard (as opposed to the Edit Wizard), you will need to create a new slideshow configuration file for the slideshow. The dialog below allows you to do so:



Initializing Wizard...

XFS v4 - Create New Slideshow

To create a new slideshow, an XML configuration file will be created. You can start with a new empty file, start with the configuration data from another slideshow, or utilize a template. Select the option you would like to use and click the Continue button.

● Start with Blank File

○ Copy Slideshow from Site: test2

○ Use Template: [Starter Template] [Demo only!] Test wi

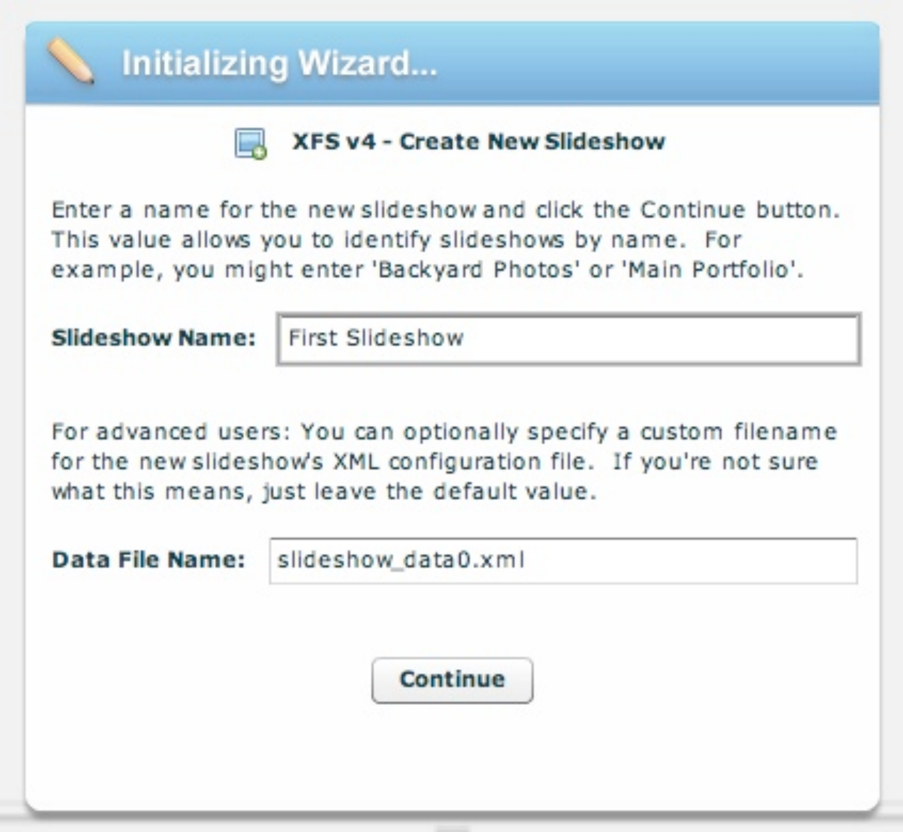
There are three different starter options for creating a new slideshow:

- You can start with a blank file. This blank file has only the most basic settings; it is essentially a *tabula rasa*.

- You can copy the configuration of another slideshow in the site. This option is only available if you have already created a slideshow in the site. The name displayed in the dropdown box will be based on the slideshow names you specified for the other slideshows (as mentioned below).
- You can use a template. When creating your very first slideshow, you'll be able to choose from the selection of starter templates. If you're created your own templates, you'll be able to choose from them as well. (For more information on using templates, see 'The Templates Menu / Using Templates'). When you hover over the starter templates, a tooltip with a preview image will be displayed.

Naming the New Slideshow

After creating the XML configuration file, you will be prompted to enter a name for the new slideshow. This allows you to identify the slideshow later, if you want to copy its configuration when creating another slideshow (as described above). This will also help you identify the data files if you are making manual edits. It is recommended that you enter a value such as 'Homepage slideshow' or 'Main portfolio.' If you don't intend to copy a slideshow, you can also enter a generic value such as 'slideshow0' or 'slideshow1':



Initializing Wizard...

XFS v4 - Create New Slideshow

Enter a name for the new slideshow and click the Continue button. This value allows you to identify slideshows by name. For example, you might enter 'Backyard Photos' or 'Main Portfolio'.

Slideshow Name: First Slideshow

For advanced users: You can optionally specify a custom filename for the new slideshow's XML configuration file. If you're not sure what this means, just leave the default value.

Data File Name: slideshow_data0.xml

Advanced Users: The Data File Name field specifies the file name which will be used for the new slideshow's XML configuration file. The default value (`slideshow_data0.xml`, `slideshow_data1.xml`, `slideshow_data2.xml`, etc) can generally be used. However, if you're managing a site with many slideshows, it may be helpful to use more meaningful file names.

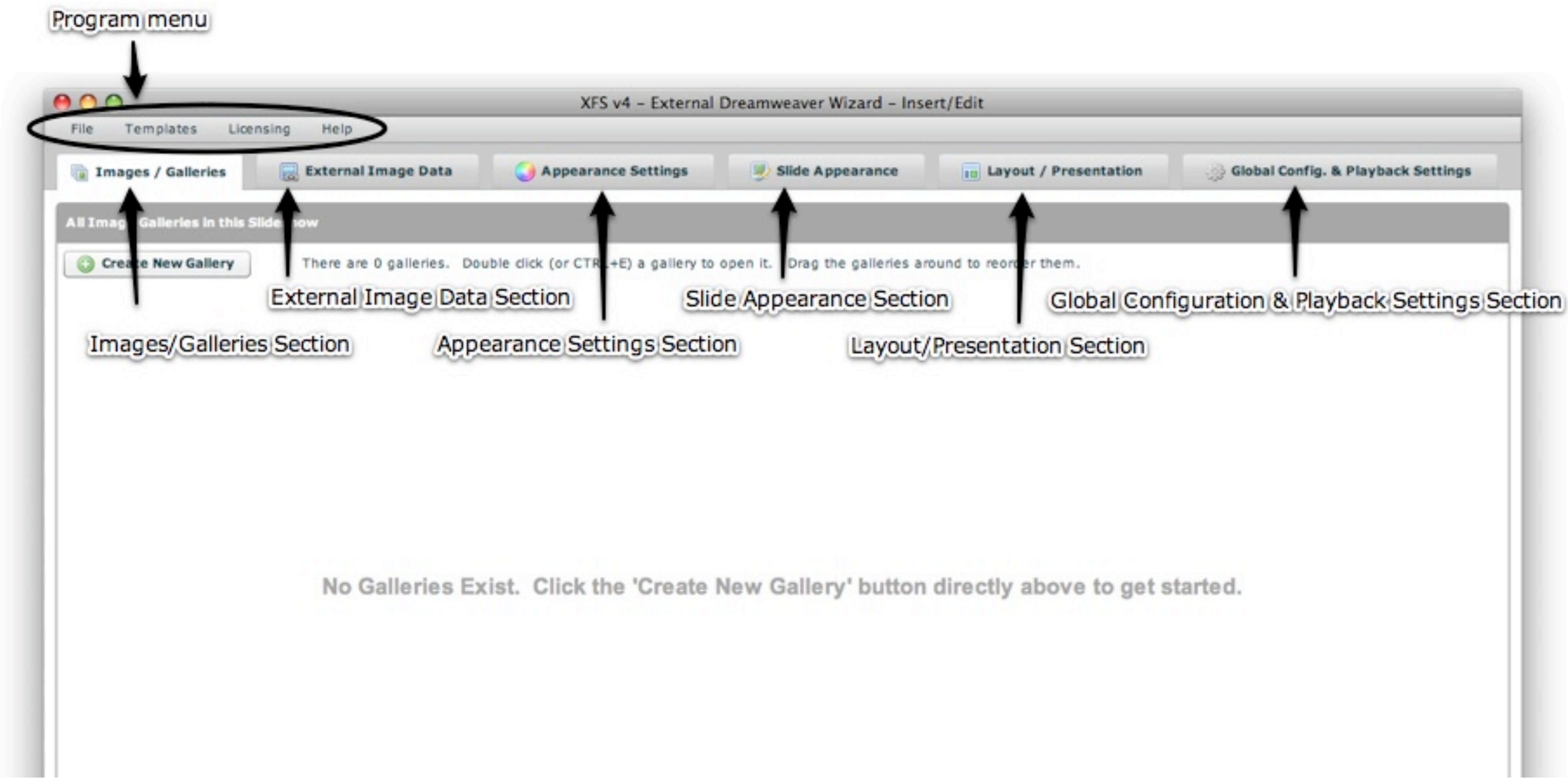
Once you have entered a name, click **Continue** to proceed to the main wizard interface.

Section-by-Section Usage Guide

This section of the manual covers each of the main aspects of the wizard. This includes other related topics, such as building layouts and templating capabilities.

Wizard Interface Overview / General Settings and Options

This section provides an overview of the wizard interface, especially general settings and options which are not slideshow-specific. The image below notes the identification names of the various sections in the wizard window:


Program menu
XFS v4 – External Dreamweaver Wizard – Insert/Edit
File
Templates
Licensing
Help
Images / Galleries
External Image Data
Appearance Settings
Slide Appearance
Layout / Presentation
Global Config. & Playback Settings
All Imag Galleries in this Slide ow
Create New Gallery
External Image Data Section
Slide Appearance Section
Global Configuration & Playback Settings Section
Images/Galleries Section
Appearance Settings Section
Layout/Presentation Section

There are 0 galleries. Double click (or CTRL+E) a gallery to open it. Drag the galleries around to reorder them.

No Galleries Exist. Click the 'Create New Gallery' button directly above to get started.

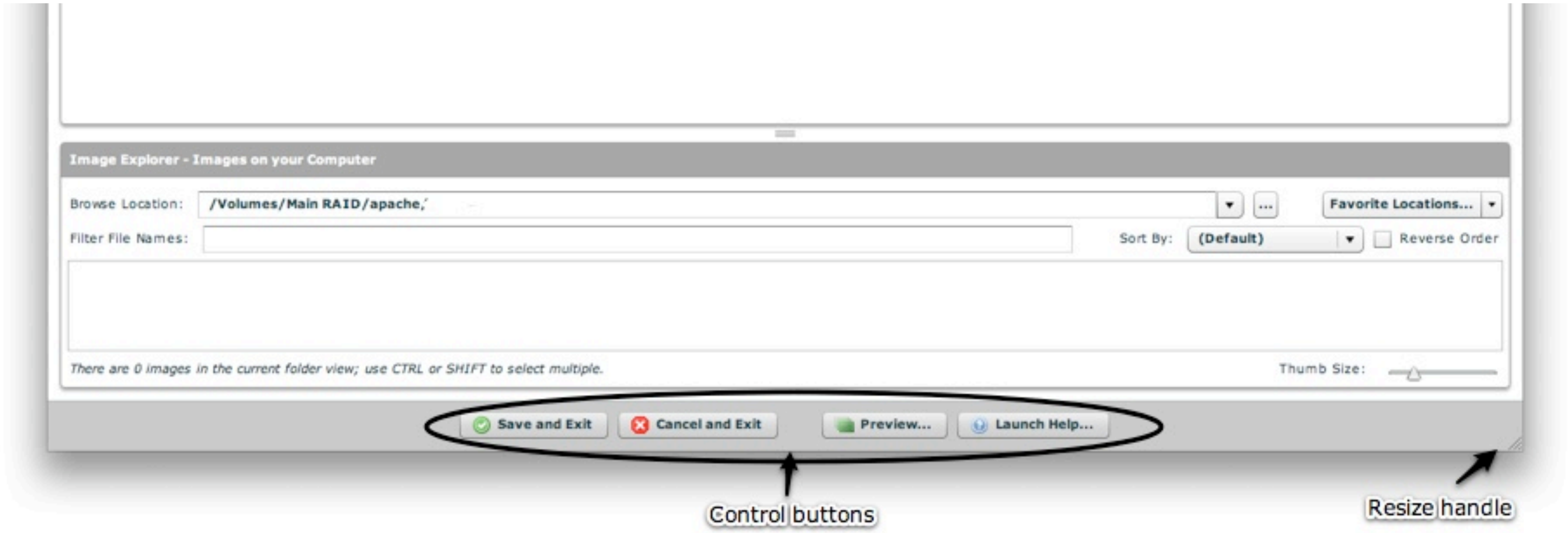


Image Explorer - Images on your Computer

Browse Location: /Volumes/Main RAID/apache,/

Filter File Names:

Sort By: (Default)

□ Reverse Order

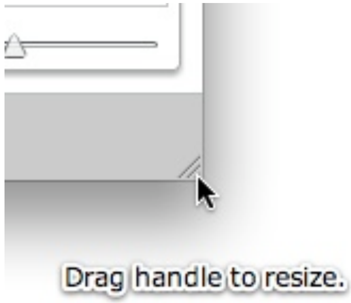
There are 0 images in the current folder view; use CTRL or SHIFT to select multiple.

Thumb Size:

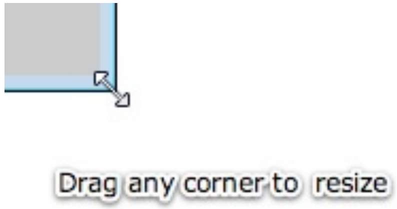
The areas covered in this section are denoted above as Program Menu and Control Buttons.

Window Management: Resizing and Repositioning

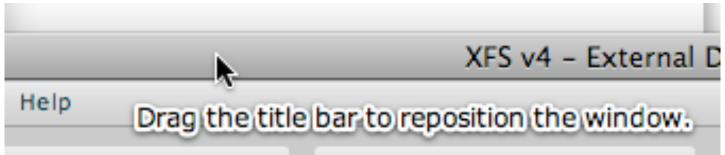
When you launch the wizard for the very first time, it will be automatically centered on your screen. You can resize the window on a Mac by dragging the resize handle in the bottom-right corner of the window:

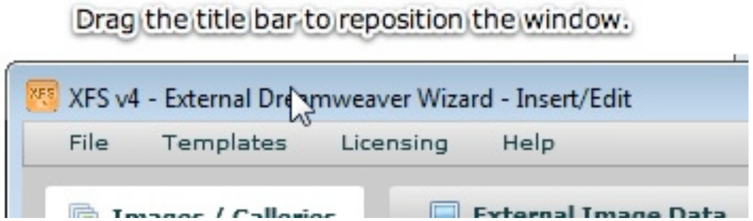


On Windows, you can resize the window by dragging any corner:



To resposition the window, drag the title bar (screenshots from Mac and Windows respectively):




Drag the title bar to reposition the window.
XFS v4 - External Dreamweaver Wizard - Insert/Edit
File
Templates
Licensing
Help
Images / Galleries
External Image Data

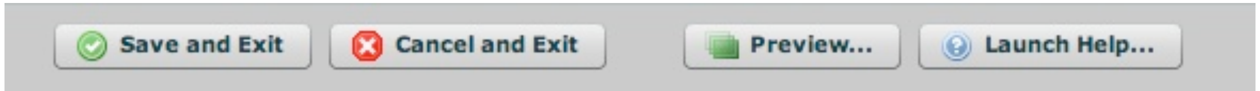
When the window is closed, the last position and size are automatically saved. On the next launch, the window will reopen in the same position and at the same size. When editing slideshows, the last-viewed tab will be automatically opened when you launch the wizard next, to help you get right back where you were.

Using the External Wizard to Create and Edit Slideshows > Section-by-Section Usage Guide > Wizard Interface Overview / General Settings and Options > **Main Buttons and Program Menu**

Main Buttons and Program Menu

Control Buttons

The main control buttons are located at the bottom of the external wizard window:

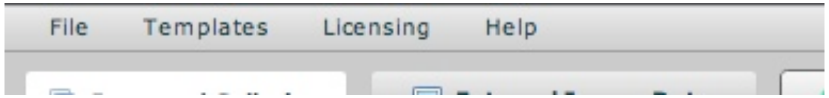


They each have the following purpose:

- **Save and Exit** - This option saves any changes you have made and closes the external wizard. If there are any problems saving the data and writing the updated configuration files, or there are any problems with the data you have entered, you will be notified so you can correct them. After using this button, you would then click the **Insert** or **Update** button in the Dreamweaver extension dialog to insert/update the slideshow HTML in your page.
- **Cancel and Exit** - This option allows you to discard any changes and closes the external wizard. A confirmation is displayed, to make sure you want to discard any changes. If you have used the **Preview...** button, you don't want to cancel and exit, because any data changes which were committed during the preview will still be retained. (More on this can be found in the Getting Started / Previewing section of the manual.) After using this button, you would then click the **Cancel** button in the Dreamweaver extension dialog to return to the main Dreamweaver window without applying any changes.
- **Preview** - This option launches the Preview window. This displays an instant preview of the slideshow. You can learn more about using the Preview feature in the Previewing Locally - In the Wizard or a Browser topic in the Getting Started section.
- **Launch Help** - This option will launch a browser window and take you to the XFS v4 support center, where you can view usage resources (such as this manual) or contact a support technician.

Program Menu

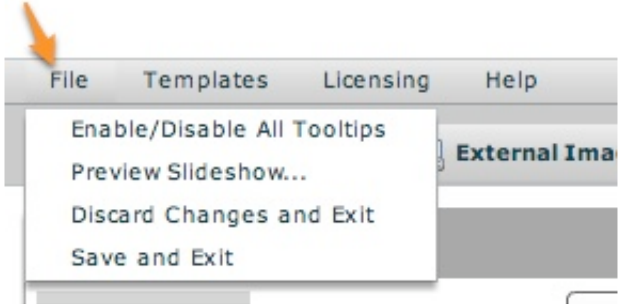
The program menu is the following menu, located at the top of the external wizard window:



This menu contains several general / global options and settings. Each individual menu is detailed in the subsequent topics.

The File Menu

The File option on the program menu appears as follows:

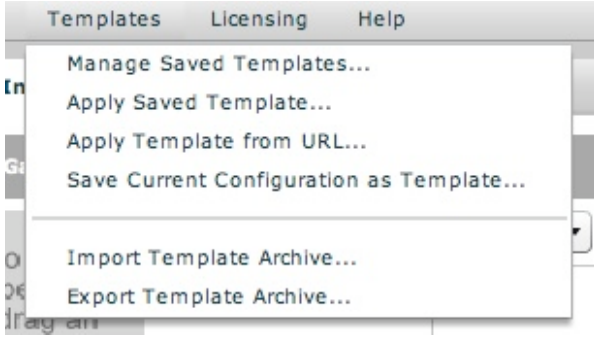


- **Enable/Disable All Tooltips** - In a number of locations throughout the wizard, various help tooltips are displayed to clarify what various options control. If you would prefer to completely disable all of these tooltips, you can toggle them with this option. **Note** - Please keep in mind that this does NOT affect any tooltips displayed in the actual slideshow; it only applies to tooltips displayed while working in the wizard.
- **Preview Slideshow...** - This option has the same effect as clicking the Preview button. It launches the preview window, allowing you to preview the slideshow.
- **Discard Changes and Exit** - This option has the same effect as clicking the Cancel and Exit button. It discards changes and closes the wizard.
- **Save and Exit** - This option has the same effect as clicking the Save and Exit program control button. It saves any changes you have made and closes the wizard.

The Templates Menu / Using Templates

Templates are a powerful feature which allows you to save, reuse and share configuration settings from a slideshow. They save all aspects of a slideshow's configuration - its layout, styles, appearance settings, playback settings, general configuration settings, as well as any special slideshow HTML parameters. When leveraged properly, this feature can save you immense amounts of time when creating multiple slideshow. This topic, along with its subtopics, details how to use this tool.

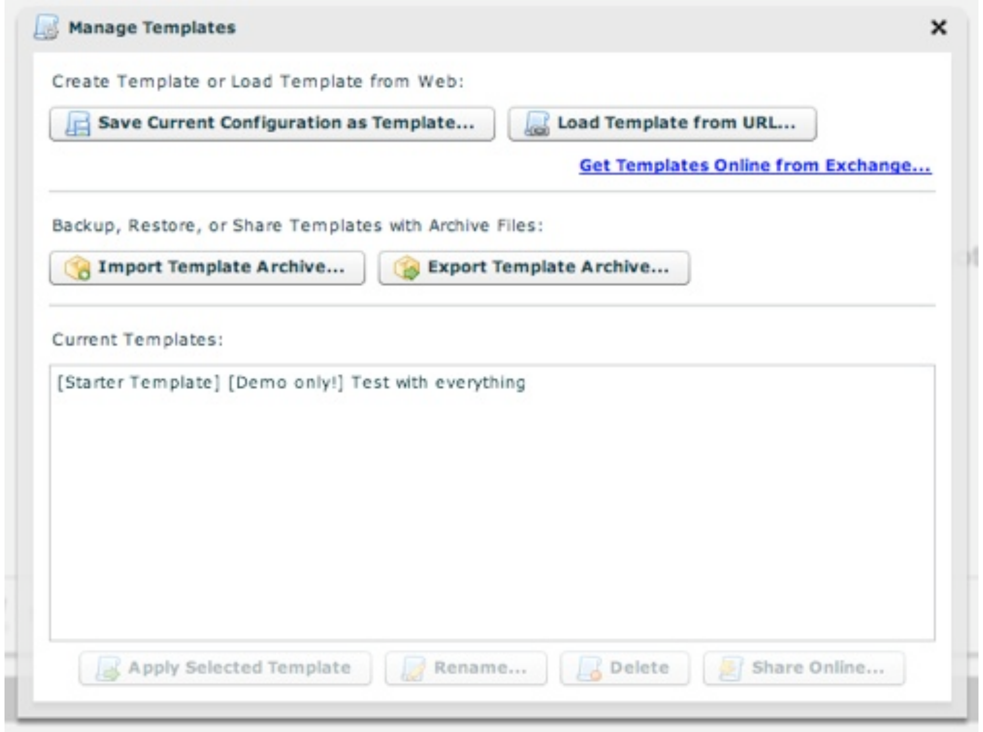
The template options can be accessed from the Templates menu. The Templates menu appears as follows:


Templates
Licensing
Help
Manage Saved Templates...
Apply Saved Template...
Apply Template from URL...
Save Current Configuration as Template...
Import Template Archive...
Export Template Archive...

- **Manage Saved Templates** - This option launches the Manage Templates dialog (described below). This is where you will manage all of your templates, add new ones, import/export archives, etc.
- **Apply Saved Template** - This option launches the Manage Templates dialog, where you can apply an existing template to your slideshow.
- **Apply Template from URL** - This option provides a shortcut to the *Load Template from URL* button in the Manage Templates dialog. It allows you to load a template directly from a URL, applying it to your slideshow and/or adding it to your template library.
- **Save Current Configuration as Template** - This option as well is a shortcut to an option in the main template management dialog. It allows you to save the current slideshow configuration as a template for later reuse.
- **Import Template Archive** - This shortcut option allows you to import a .xst template archive file, and add the templates contained in it to your library. See the Template Archives subtopic for more information.
- **Export Template Archive** - This shortcut option allows you to export your current templates to a .xst template archive file. See the Template Archives subtopic for more information.

Manage Templates Dialog

The Manage Templates dialog (accessed by Templates > Manage Saved Templates...) offers access to all of the template managment options:


Manage Templates
✕
Create Template or Load Template from Web:
Save Current Configuration as Template...
Load Template from URL...
Get Templates Online from Exchange...
Backup, Restore, or Share Templates with Archive Files:
Import Template Archive...
Export Template Archive...
Current Templates:
Apply Selected Template
Rename...
Delete
Share Online...

[Starter Template] [Demo only!] Test with everything

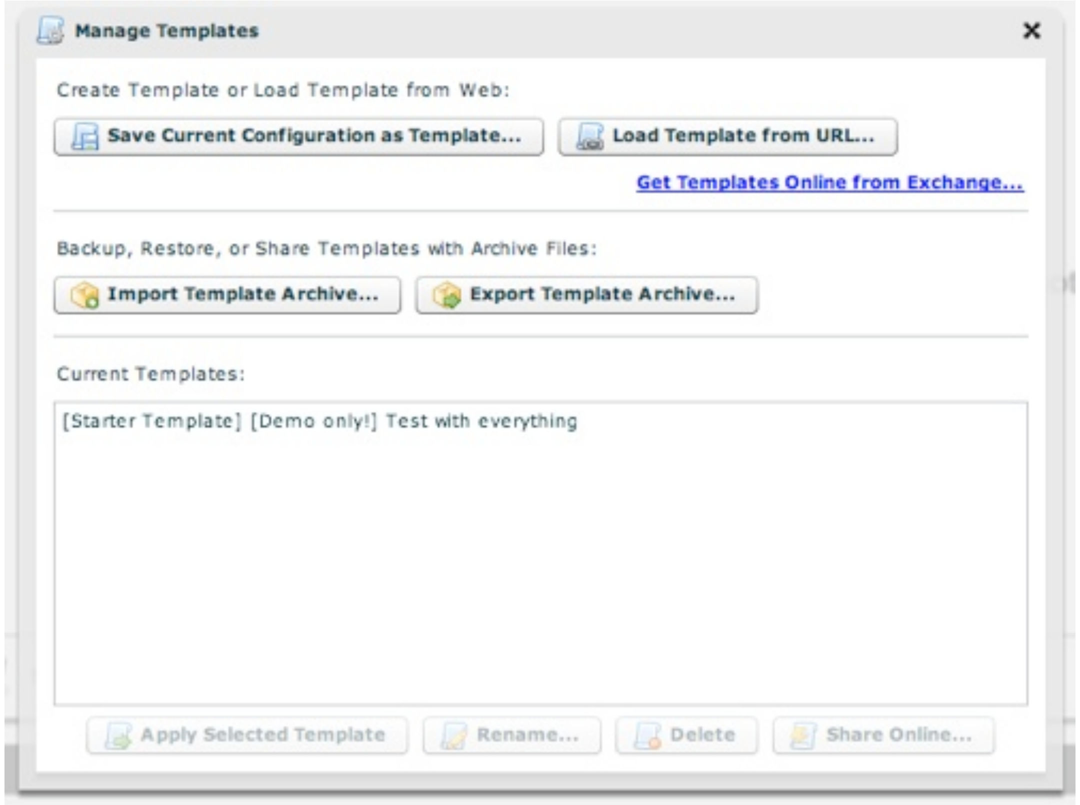
All of the options in this dialog are covered in the following subtopics.   To close the dialog, click the 'X' in the upper-right corner.

Note: The screenshots in this section don't include the latest preview feature. When you select a starter template in the Current Templates list, a preview image will be shown to the right of the list.

Basic Template Techniques

This section covers the following options in the Manage Templates dialog:

- Saving a slideshow's configuration as a template
- Managing existing templates
- Applying a template from your library to a slideshow



Manage Templates
Create Template or Load Template from Web:
Save Current Configuration as Template...
Load Template from URL...
Get Templates Online from Exchange...
Backup, Restore, or Share Templates with Archive Files:
Import Template Archive...
Export Template Archive...
Current Templates:
Apply Selected Template
Rename...
Delete
Share Online...

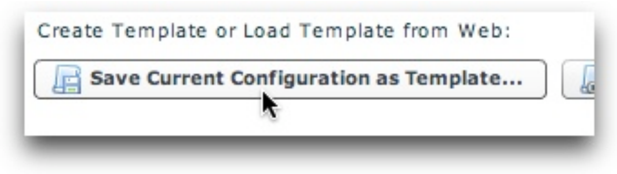
[Starter Template] [Demo only!] Test with everything

Note: You will likely have many more starter templates than shown in the screenshots. This demo doesn't show all of these templates.

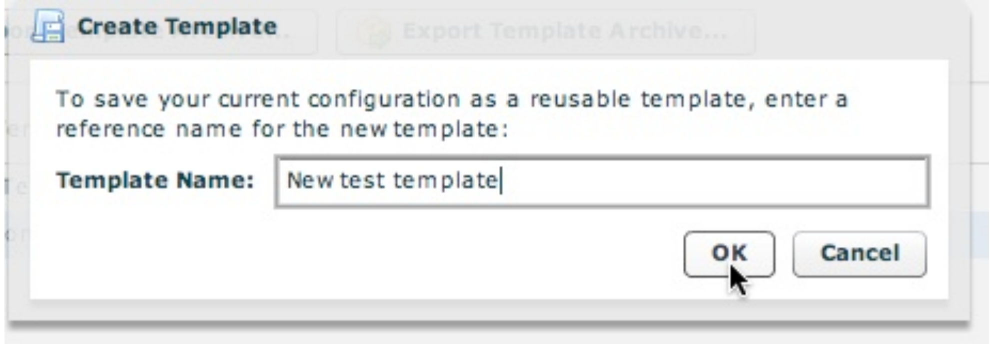
Tip: To learn more about all of the available starter templates, see the Starter Templates page.

Saving a Slideshow's Configuration as a Template

To reuse a slideshow's configuration at a later time, you need to save it as a template. To do so, click the **Save Current Configuration as Template...** button:


Create Template or Load Template from Web:
Save Current Configuration as Template...

You will be prompted to enter a name for the new template:



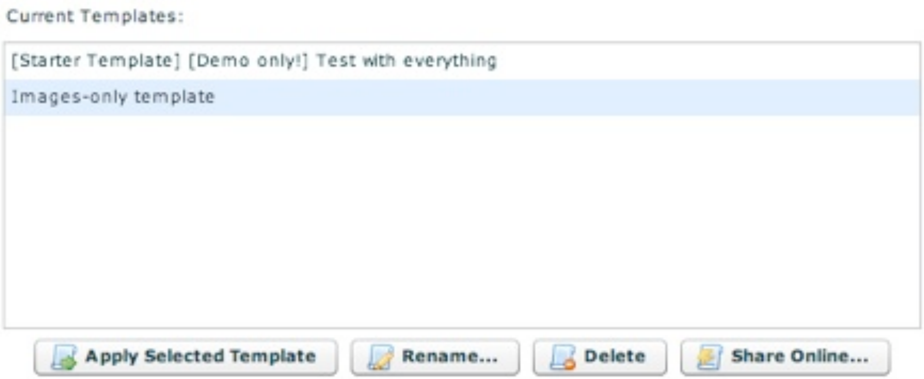
To save your current configuration as a reusable template, enter a reference name for the new template:

Template Name: New test template

Once you have created the template, it will appear in your Current Templates list.

Managing Existing Templates

All of the templates in your library are listed in the Current Templates list (you will likely have many more than shown in this example):

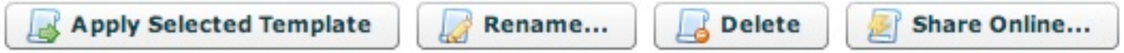


Current Templates:

[Starter Template] [Demo only!] Test with everything

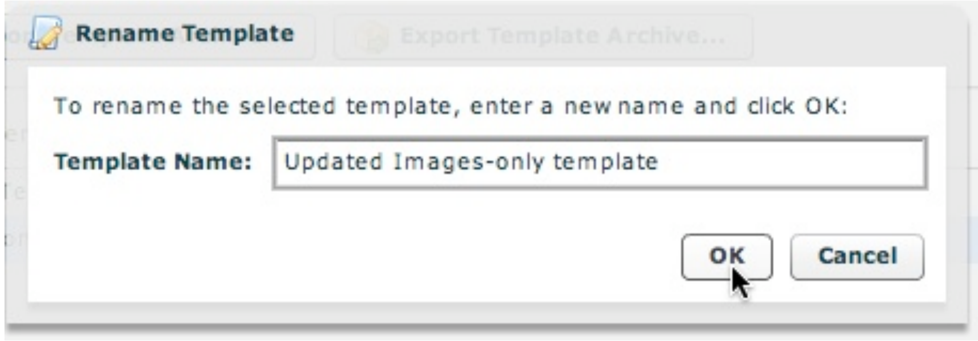
Images-only template

Once you select a template, there are several management buttons which become active:



Apply Selected Template - This button, as described in more detail below, allows you to apply the selected template to your current slideshow. Keep in mind that the existing configuration will be overwritten.

Rename - This button allows you to rename the template. You will receive a prompt which allows you to enter a new name and save the changes:



To rename the selected template, enter a new name and click OK:

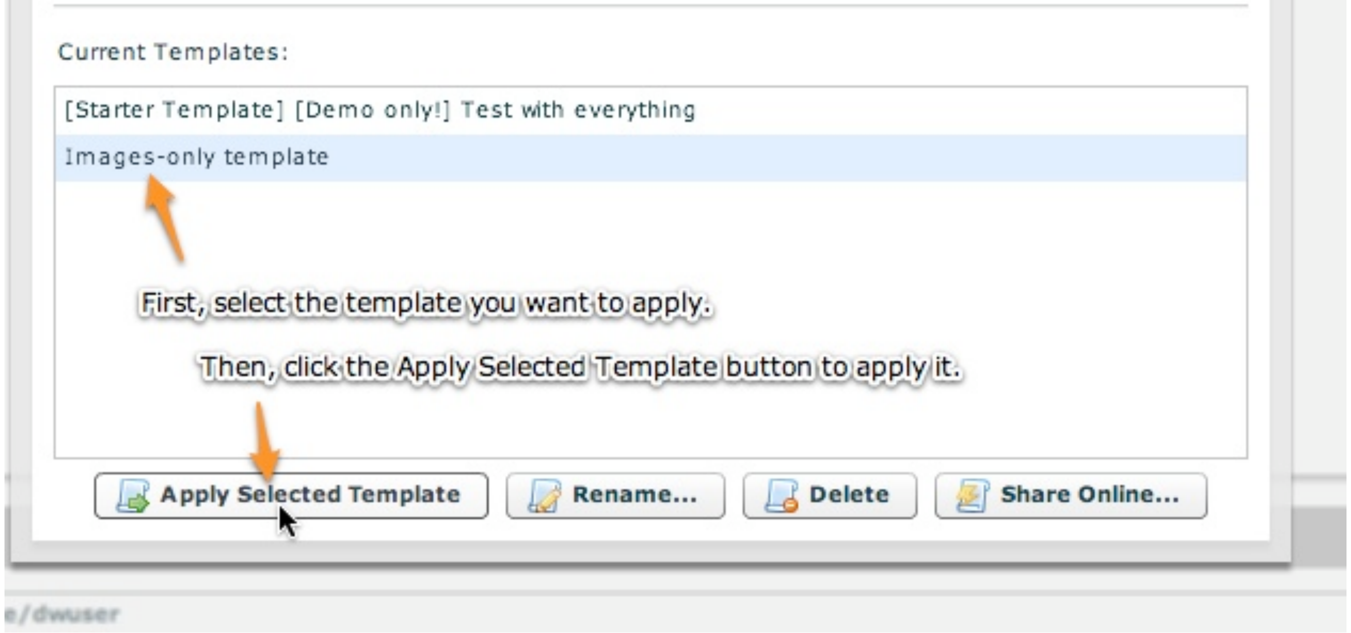
Template Name: Updated Images-only template

Delete - This button allows you to permanently remove a template from your library. A confirmation dialog is displayed, to ensure that you don't accidentally remove any templates. No slideshows which used the template are affected by removing a template; it is only removed from your template library.

Share Online - Allows you to submit a template to the template exchange. For additional information, see the Getting and Sharing Templates Online topic.

Applying a Template from Your Library To a Slideshow

There are two points at which you can apply a template to a slideshow. When first creating the slideshow, you can select a template as the starting point in the New Slideshow dialog. This is described in the Creating a New Configuration File topic. You can also apply a template to an already-created slideshow by using the Manage Templates dialog. To do so, select the template you want to apply from the Current Templates list, then click the **Apply Selected Template** button:



First, select the template you want to apply.

Then, click the Apply Selected Template button to apply it.

The template will be applied and you will be notified. Remember that any existing settings will be overwritten when you apply a template. Because of this, you may want to save your existing configuration as a backup template before applying the new template. This way, if something goes wrong, you can revert to the old configuration.

Getting and Sharing Templates Online

Once you install the software, your list of templates doesn't have to become stale. Instead, there are several online options to help you continue to expand creativity.

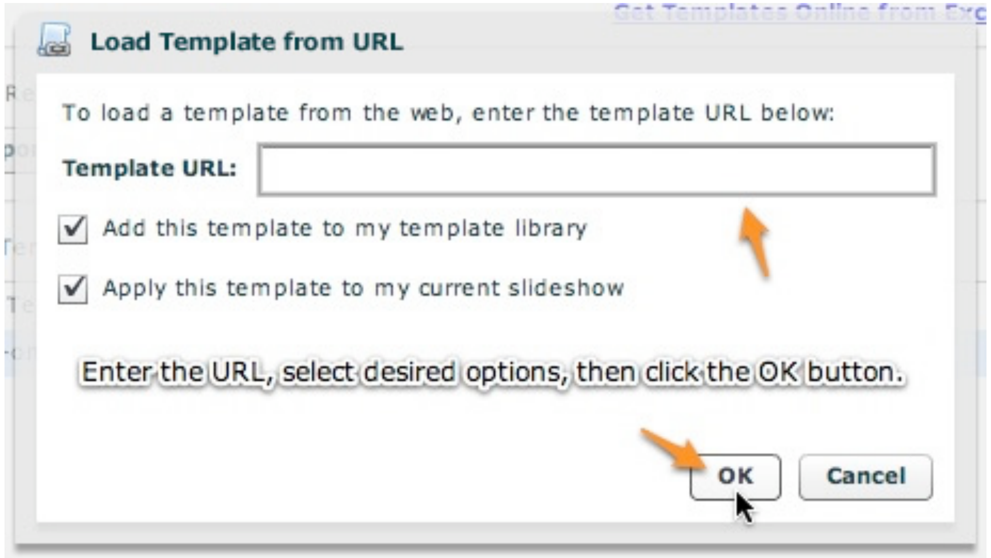
- Occasionally, new starter templates will be automatically downloaded and installed into the program when it checks for updates.
- The XFS v4 Template Exchange offers a central repository where XFS v4 users can share their work.   You can upload templates to share, or browse and add new templates to your library.
- The Starter Templates page shows all of the starter templates which are available. These will likely already be installed ino your template library, but you can also grab the template URL directly from the page.
- If you find a slideshow on the web you want to duplicate, you can ask the publisher for the configuration file URL and their permission to use it on your slideshows.
- New DWUser.com-created template packs may be made available in the future for free download, or purchased for a nominal fee.   See the main XFS v4 product page for more information.

Within the wizard, you will use the following methods to upload or download templates.

Getting Additional Templates

To get a template from the web, first obtain the template URL using one of the above methods. Next, in the Manage Templates dialog, click the **Load Template from URL** button (or select Templates > Apply Template from URL...):

In the dialog box which opens, specify the URL from which the template should be loaded.   By selecting or deselecting the option checkboxes, you can specify if you want to only apply the template to your current slideshow, only add it to your library, or both:



Load Template from URL

To load a template from the web, enter the template URL below:

Template URL:

☑ Add this template to my template library

☑ Apply this template to my current slideshow

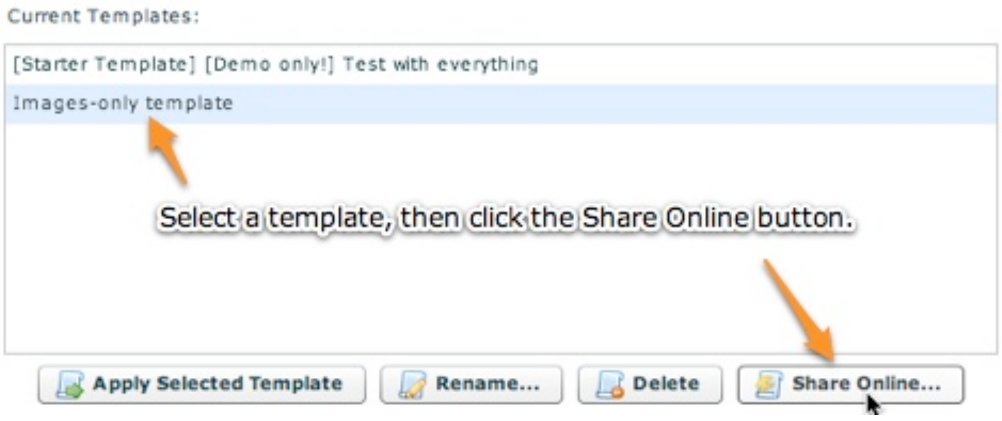
Enter the URL, select desired options, then click the OK button.

Once you've filled in the options, click the OK button. The template will load and you will be notified once the loading process has completed.

Note: Some online templates, such as template packs, may be distributed as downloadable template archive files. To learn more about using template archives, see the Template Archives page.

Sharing / Uploading Templates

Created a great template? You can upload it to the Template Exchange where other users can download and use it. To do this, select a template in the Current Templates list and click the **Share Online** button:



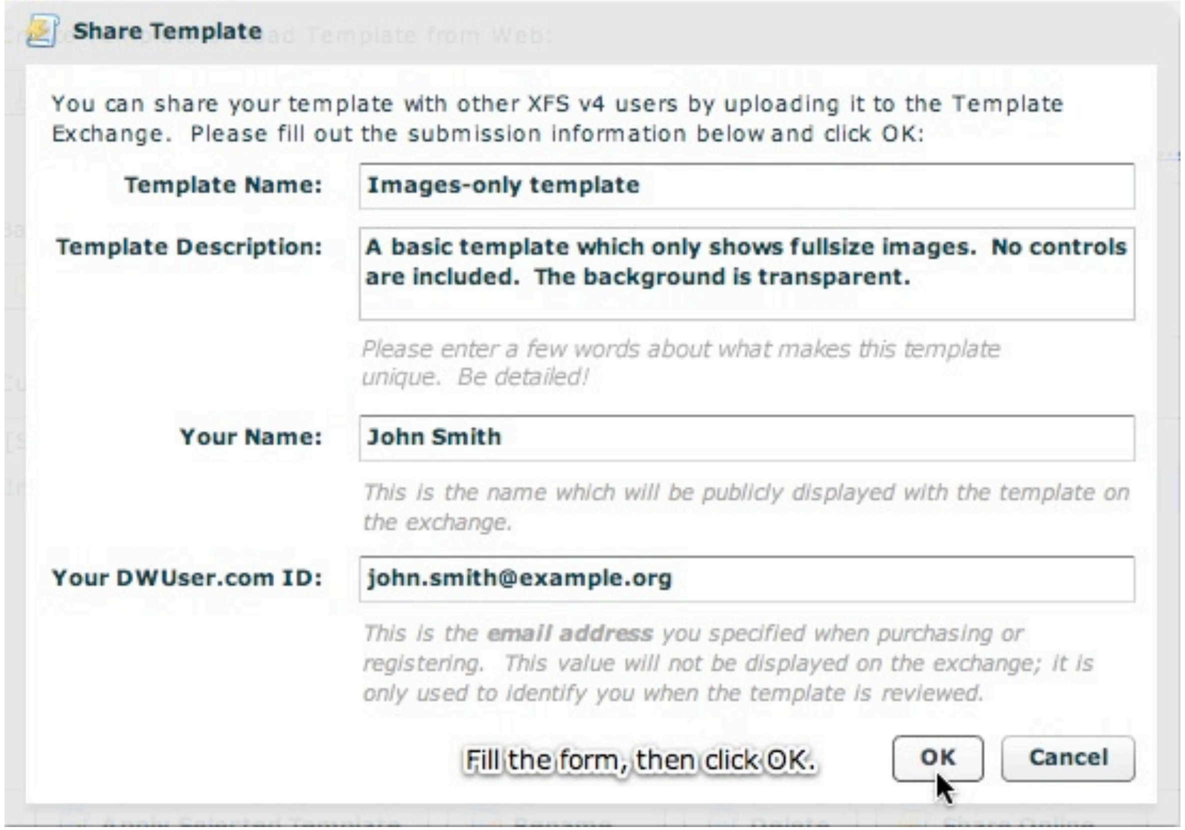
Current Templates:

[Starter Template] [Demo only!] Test with everything

Images-only template

Select a template, then click the Share Online button.

The Share Template dialog will open:



You can share your template with other XFS v4 users by uploading it to the Template Exchange. Please fill out the submission information below and click OK:

Template Name:	Images-only template
Template Description:	A basic template which only shows fullsize images. No controls are included. The background is transparent.

Please enter a few words about what makes this template unique. Be detailed!

Your Name:	John Smith

This is the name which will be publicly displayed with the template on the exchange.

Your DWUser.com ID:	john.smith@example.org

This is the ***email address*** *you specified when purchasing or registering. This value will not be displayed on the exchange; it is only used to identify you when the template is reviewed.*

Fill the form, then click OK.

There are several parameters which you need to provide:

- **Template Name** - This value is automatically filled with the template's name. If the default name doesn't give a concise idea of what the template is about, add a few more words.
- **Template Description** - This should describe in detail what makes the template unique. Verbosity is good - help other users have enough information to evaluate if the template is applicable for their needs.
- **Your Name** - This is the name which will be publicly associated with the template on the exchange.
- **Your DWUser.com ID** - This is the email address you used when registering or purchasing a product.   This value is **not** publicly displayed; it is only used to identify you when QA reviews the template.

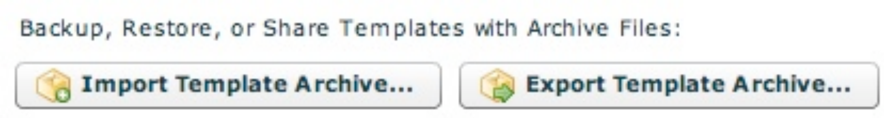
Once you're finished, click OK. The template will be uploaded and you will be notified of the successful submission.

Using the External Wizard to Create and Edit Slideshows > Section-by-Section Usage Guide > Wizard Interface Overview / General Settings and Options > The Templates Menu / Using Templates > **Template Archives**

Template Archives

All of the template management features to this point have been related to single templates. This section covers Template Archive files. These files, which end in .xst, are special archive files which can hold one or more template files. These archives can be uploaded, downloaded, shared, or used for backups. Template archive files are also used to distribute template packs - pre-built packages of multiple templates.

In the template management dialog, there are two archive-related options - Import Template Archive and Export Template Archive:

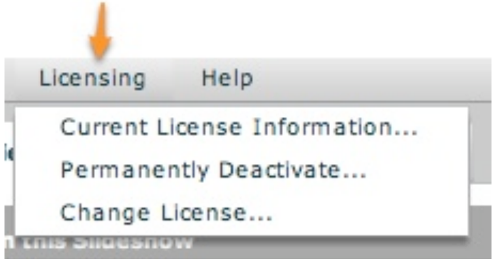

Backup, Restore, or Share Templates with Archive Files:
Import Template Archive...
Export Template Archive...

The **Export Template Archive** option allows you to export all of the templates in your library as an XST template archive. You will be prompted to select a location to save the file to, then the archive will be generated and written to disk. You can use this option to share all of your templates, create a backup, or transfer your templates to another computer.

The **Import Template Archive** option will prompt you to select an existing XST template archive on your computer.   Once you select one, the templates will be imported and added to your existing library. This can be used to restore all of your templates from another installation or another computer.

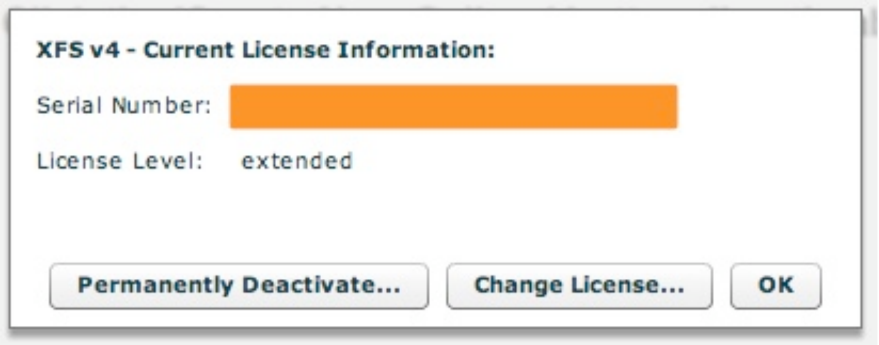
Licensing Menu

The Licensing menu allows you to manage product licensing. The menu appears as follows:



Current License Information Dialog

The Current License Information option displays information about your current license:


XFS v4 - Current License Information:

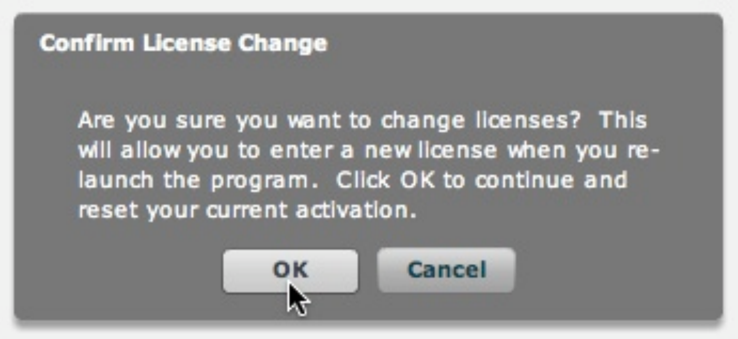
Serial Number: [redacted]

License Level: extended

The License Level value specifies if you have the Basic (*basic*) or Pro (*extended*) version activated based on the license type.

Changing Licenses

If you upgrade to the Pro version of the software, you must enter the new serial number to activate the Pro version. To do this, select the Change License option. You will receive this message:



Confirm License Change

Are you sure you want to change licenses? This will allow you to enter a new license when you re-launch the program. Click OK to continue and reset your current activation.

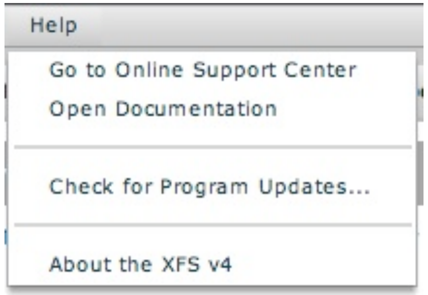
Click OK to continue. Your license will be removed, and the next time you re-launch the program you will be prompted to re-activate. You can enter the new Pro version serial number at that time, and all Pro features will be enabled.

Permanent License Deacivation

The Permanently Deactivate option allows you to destroy the license so that it is no longer usable. This option should only be used if you are returning the software for a refund and a support technician directs you to do so. You will be prompted to confirm this action, the deactivation will take place, then the program will close.

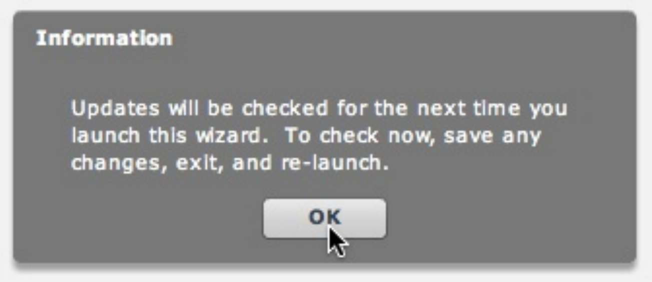
Help Menu

The Help Menu appears as follows:



Each option does the following:

- **Go to Online Support Center** - Opens a browser window and directs you to the XFS v4 Support Center.
- **Open Documentation** - Provides a shortcut to view this documentation online.
- **Check for Program Updates** - By default, the program periodically checks for updates and any new starter templates which might be available. However, you can force an immediate check for updates by selecting this option. You will receive this message:

  Updates will be checked for the next time you launch this wizard. To check now, save any changes, exit, and re-launch.

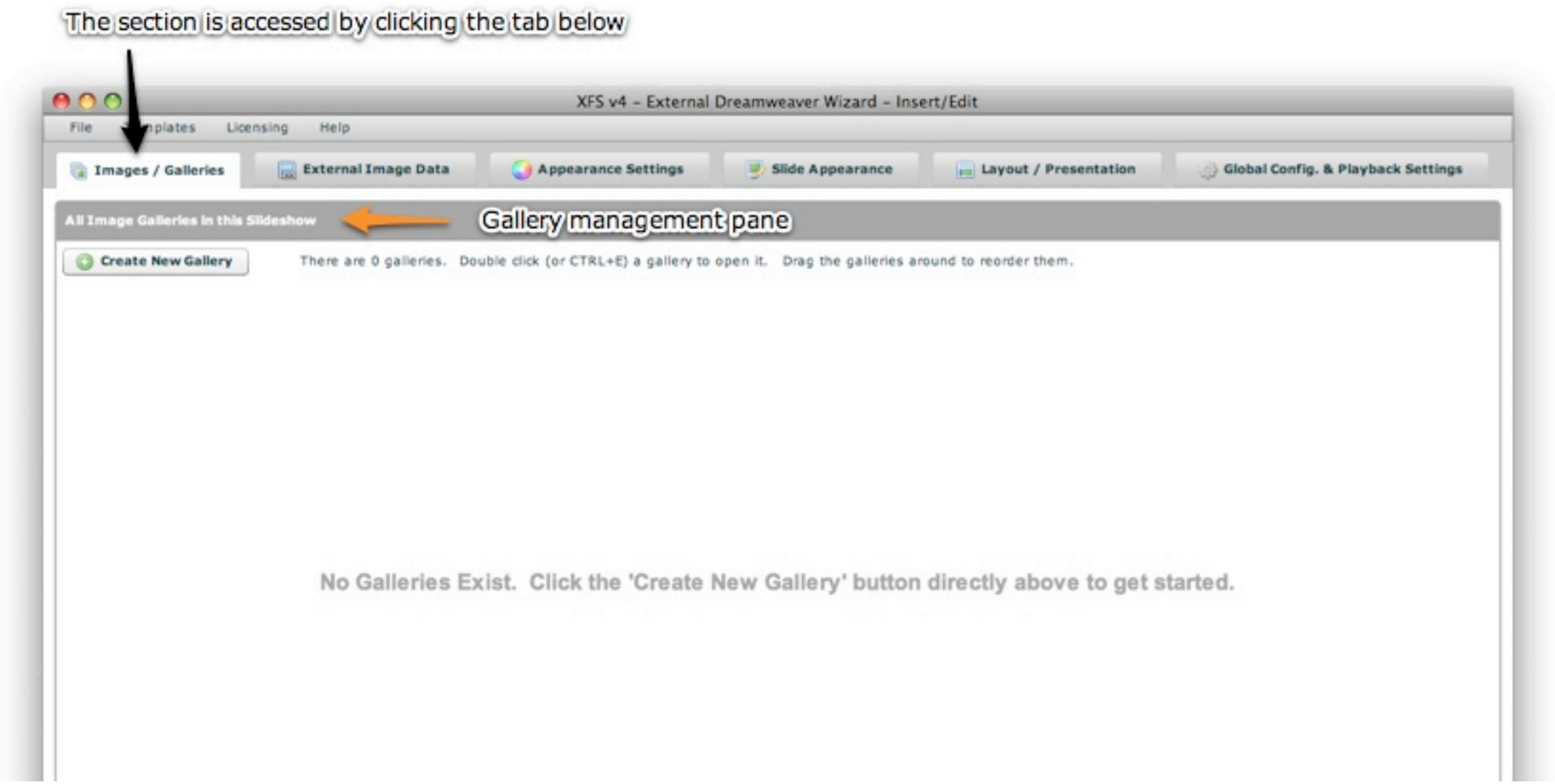
  The next time you launch the wizard, program updates will be checked for.
- **About the XFS v4** - Displays a dialog with program information, including the build version.

Using the External Wizard to Create and Edit Slideshows > Section-by-Section Usage Guide > **Using the 'Images / Galleries' Section**

Using the 'Images / Galleries' Section

The **Images / Galleries** section of the wizard is perhaps the most important section. In it, you specify the galleries which will be displayed within your slideshow along with the images in these galleries. The section is accessed by selecting the Images / Galleries tab at the top of the wizard window. If you haven't already, it is recommend you review the Definitions page before continuing, so you'll understand more clearly what each option represents.

At a glance, the section will look like this when you first create a slideshow:

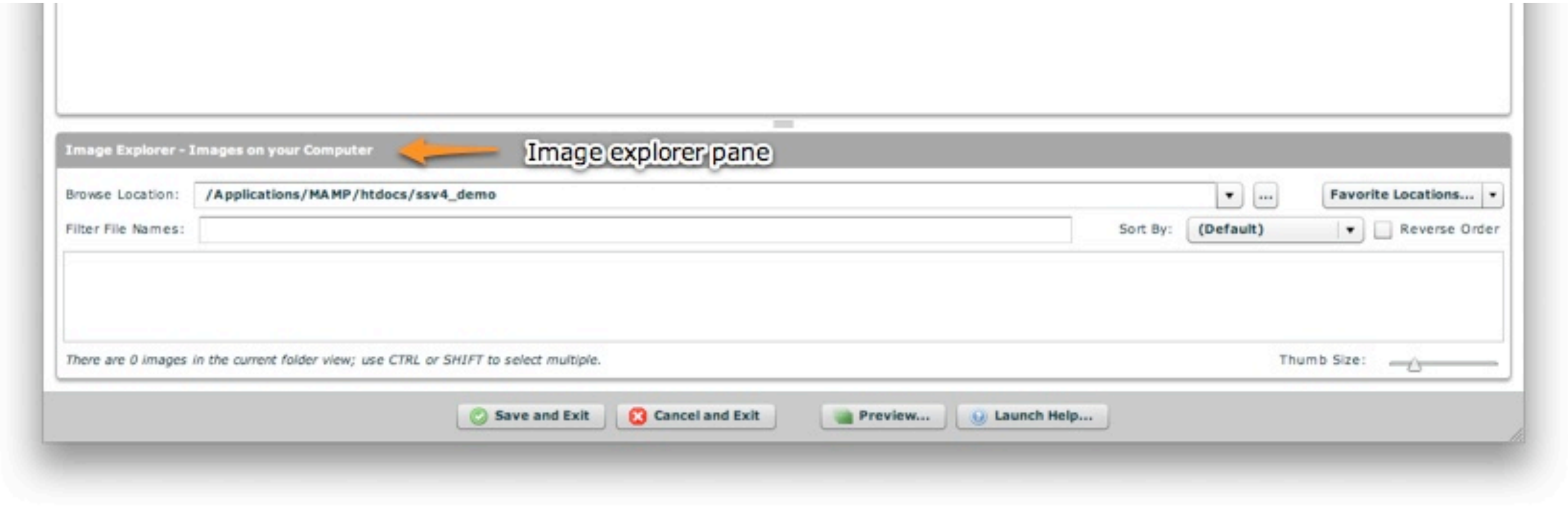


The section is accessed by clicking the tab below

Gallery management pane

There are 0 galleries. Double click (or CTRL+E) a gallery to open it. Drag the galleries around to reorder them.

No Galleries Exist. Click the 'Create New Gallery' button directly above to get started.



The **Gallery Management Pane** is where you create and manage the galleries and their images. The **Image Explorer Pane** is where you find images to add to your slideshow.

This section has several subtopics which cover usage in detail. First, you'll learn how to use the Image Explorer Pane to locate images. Then, you'll learn how to use the Gallery Management Pane to create galleries and add to them the images you selected earlier. Finally, you'll learn in detail about the various options for each slide.

Using the External Wizard to Create and Edit Slideshows > Section-by-Section Usage Guide > Using the 'Images / Galleries' Section > **The Image Explorer Pane**

The Image Explorer Pane

Located at the bottom of the wizard window, the Image Explorer Pane is where you find images to add to your slideshow. You select a folder, a preview of its images is shown, then you drag the images into a gallery. The pane appears as follows:

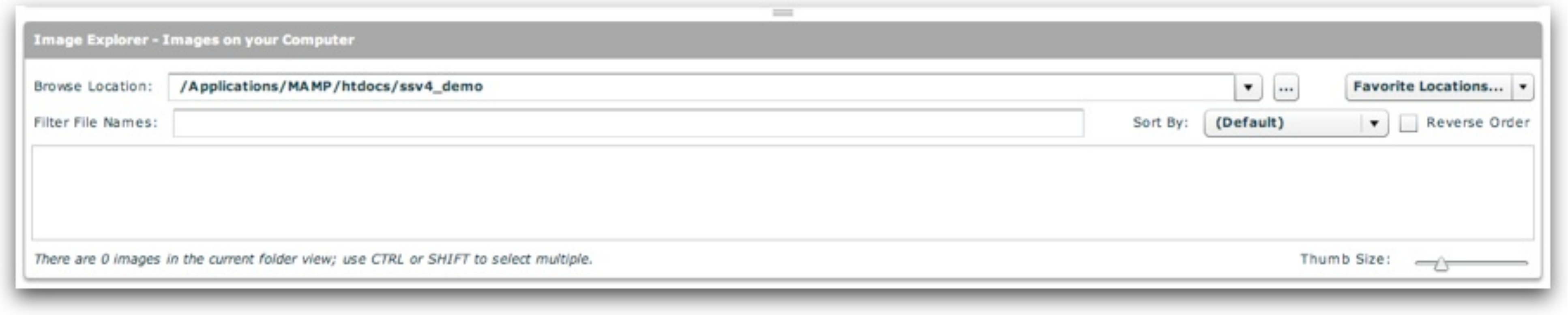


Image Explorer - Images on your Computer

Browse Location: /Applications/MAMP/htdocs/ssv4_demo

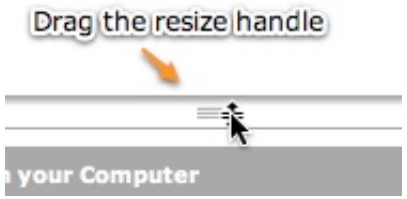
Filter File Names:

Sort By: (Default)

□ Reverse Order

There are 0 images in the current folder view; use CTRL or SHIFT to select multiple.

To resize the height of the pane (or hide it entirely), drag the resize handle at its top:



Pane Anatomy

The **Browse Location** dropdown menu allows you to specify the folder in which to look for images. By default, the site root is shown in the browse box. More on this is detailed below.

The **Favorite Locations** dropdown menu allows you to store your favorite browse locations for easier access. More on this is detailed below.

The **Filter File Names** box allows you to only show some of the images in the folder, requiring that their file names contain the specified filter string.

The **Sort By** dropdown menu allows you to change the default sort order.

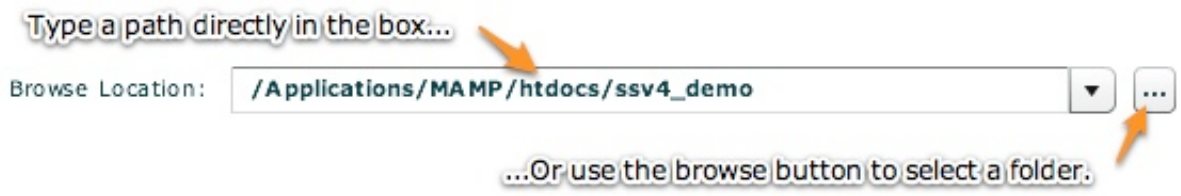
The **Reverse Order** checkbox allows you to reverse the sort defined by Sort By.

The **Main Box** displays a preview of all of the usable images (JPEGs, PNGs, GIFs, and SWFs) in the selected folder, filtered and sorted based on the other options.

The **Thumb Size** slider allows you to change the size of the thumbnail preview images in the main box.

Using the Pane - Selecting a Folder

First, you must select a folder to use. You can type a path manually or use the browse button to jump straight to a specific folder:



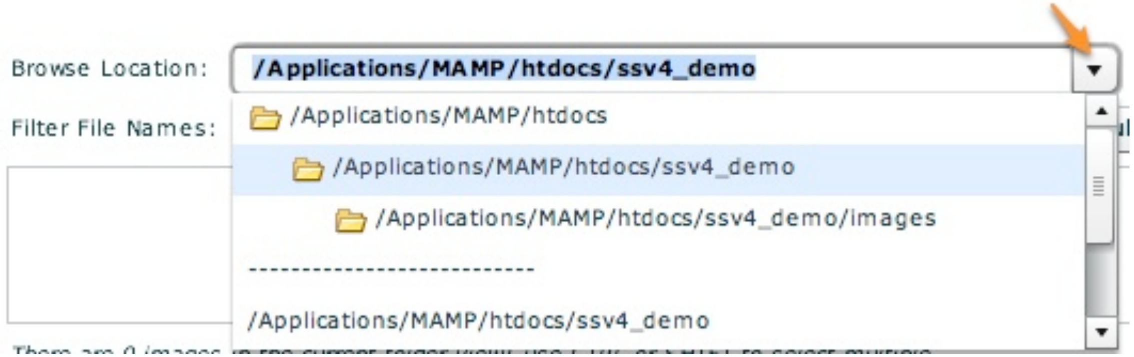
Type a path directly in the box...

Browse Location: /Applications/MAMP/htdocs/ssv4_demo

...Or use the browse button to select a folder.

Note: You must use an image folder that is **within your Dreamweaver site**. Otherwise, the images won't be accessible to your slideshow when you deploy the slideshow.   However, there is no requirement for the specific location of the images within the site - they can be placed anywhere.

You can also jump to the parent directory or any child directory by using the dropdown arrow on the right edge of the box:

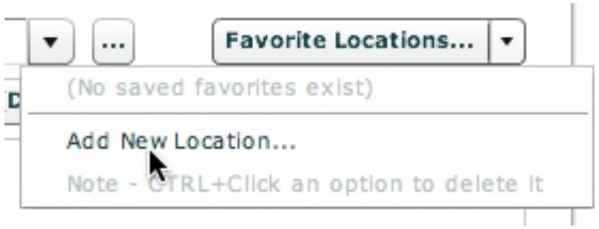

Browse Location:
/Applications/MAMP/htdocs/ssv4_demo
Filter File Names:
/Applications/MAMP/htdocs
/Applications/MAMP/htdocs/ssv4_demo
/Applications/MAMP/htdocs/ssv4_demo/images

/Applications/MAMP/htdocs/ssv4_demo

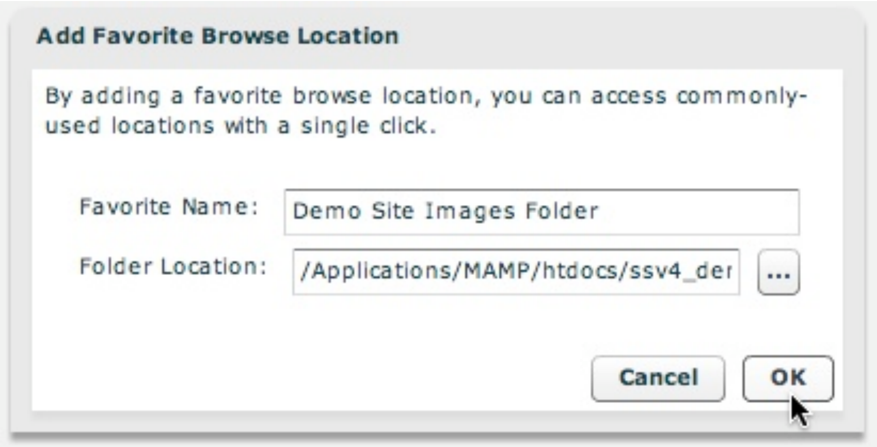
The list displays the current path's parent directory and all of its child directories. At the bottom of the list, all of your recent browse locations are displayed. The last browsing location is automatically remembered (separately for each site) and restored when you re-launch the wizard.

Using the Pane - The Favorite Locations menu

You can also save a specific folder as a favorite location, allowing you to jump straight to the folder by selecting it from the favorites menu. To add a new item, select the Add New Location option from the dropdown:


...
Favorite Locations...
(No saved favorites exist)
Add New Location...
Note - CTRL+Click an option to delete it

In the dialog which appears, specify the location of the folder and the name you want displayed in the Favorite Locations menu, then click OK:



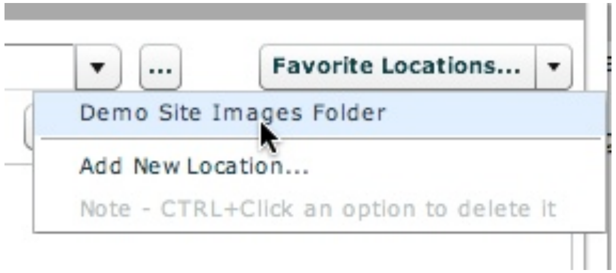
Add Favorite Browse Location

By adding a favorite browse location, you can access commonly-used locations with a single click.

Favorite Name: Demo Site Images Folder

Folder Location: /Applications/MAMP/htdocs/ssv4_der

The location will now appear in the menu:



Note - CTRL+Click an option to delete it

Click on a saved location to select it in the Browse Location box. To **remove a favorite location**, CTRL+click (CMD+click on Macintosh) the option and it will be removed.

Using the Pane - Viewing and Filtering Images

Once you have selected a folder which contains usable images, they will appear in the main box:

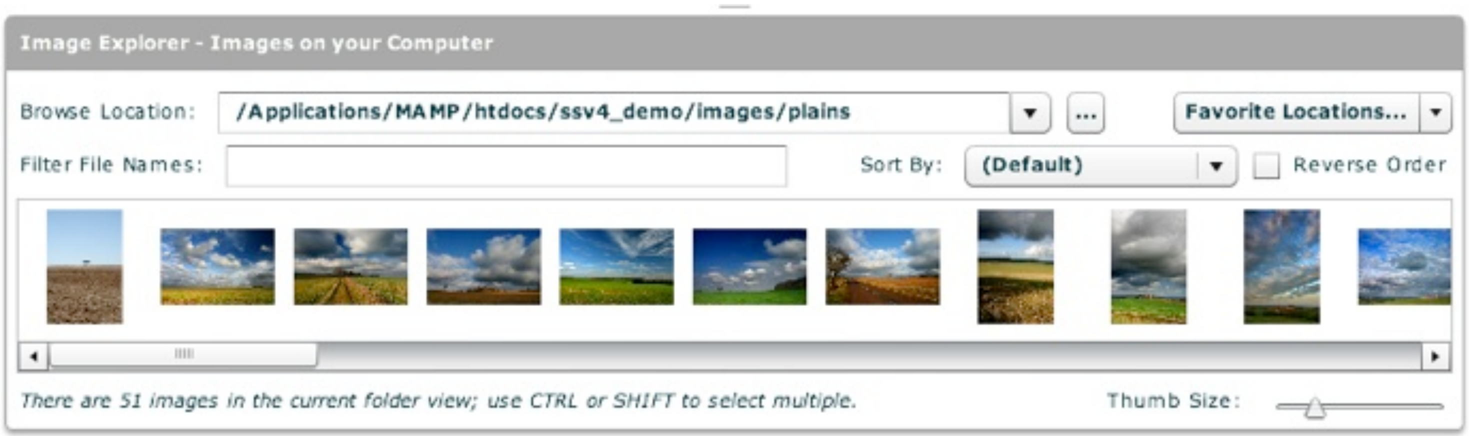


Image Explorer - Images on your Computer

Browse Location: /Applications/MAMP/htdocs/ssv4_demo/images/plains

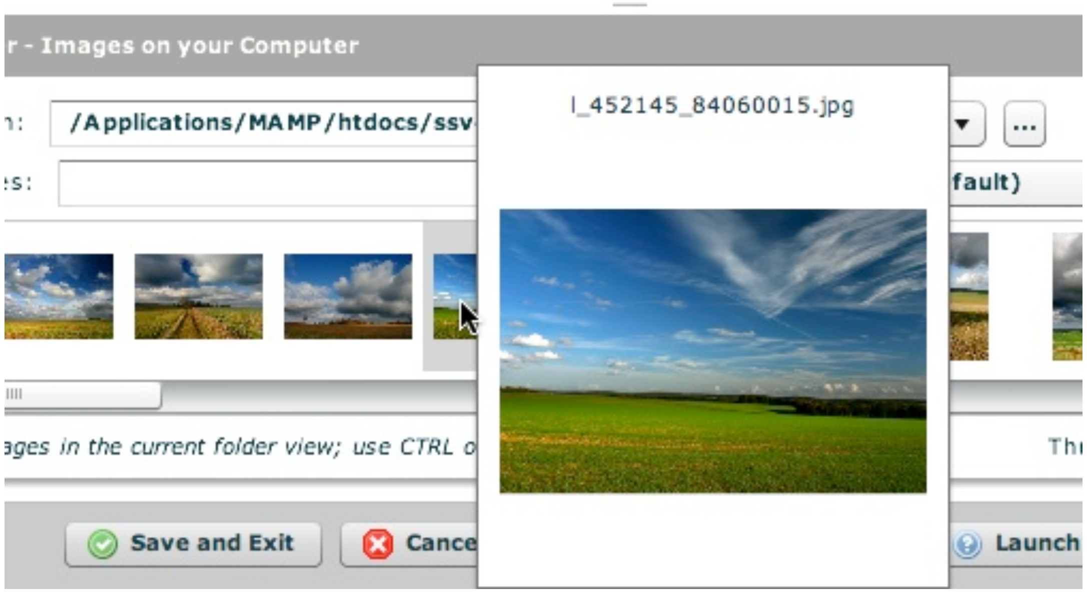
Filter File Names:

Sort By: (Default)

□ Reverse Order

There are 51 images in the current folder view; use CTRL or SHIFT to select multiple.

You can hover over an image to view its filename, dimensions, and a larger preview image: *(Note - this screenshot does not include the dimensions; this feature was added in a later build.)*



r - Images on your Computer

n: /Applications/MAMP/htdocs/ssv

es:

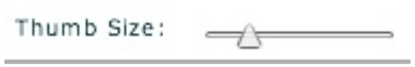
fault)

l_452145_84060015.jpg

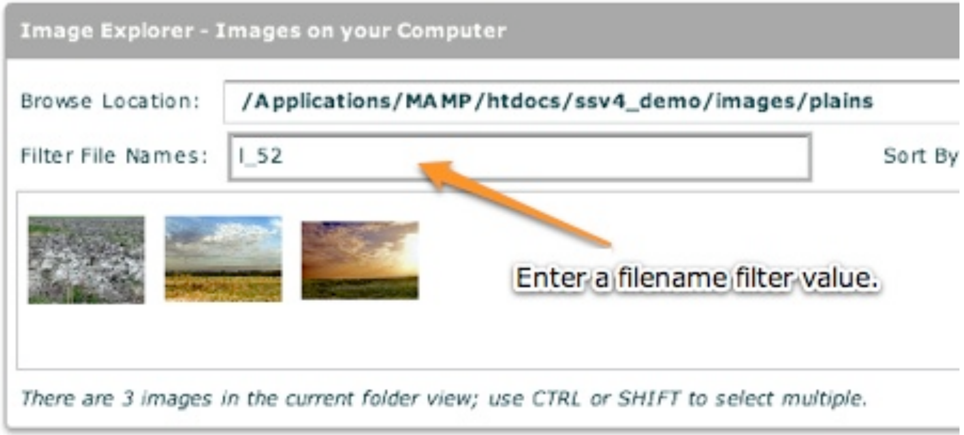
ages in the current folder view; use CTRL o

Tip: If you don't want these tooltips to appear, you can select File > Enable/Disable All Tooltips to prevent them from appearing.

To change the size of the thumbs in the preview box, drag the Thumb Size slider directly below it:



To filter the images shown by their filename, you can enter a value which must appear in the filename for the image to be included in the list. The value can appear anywhere in the filename (not just the beginning):



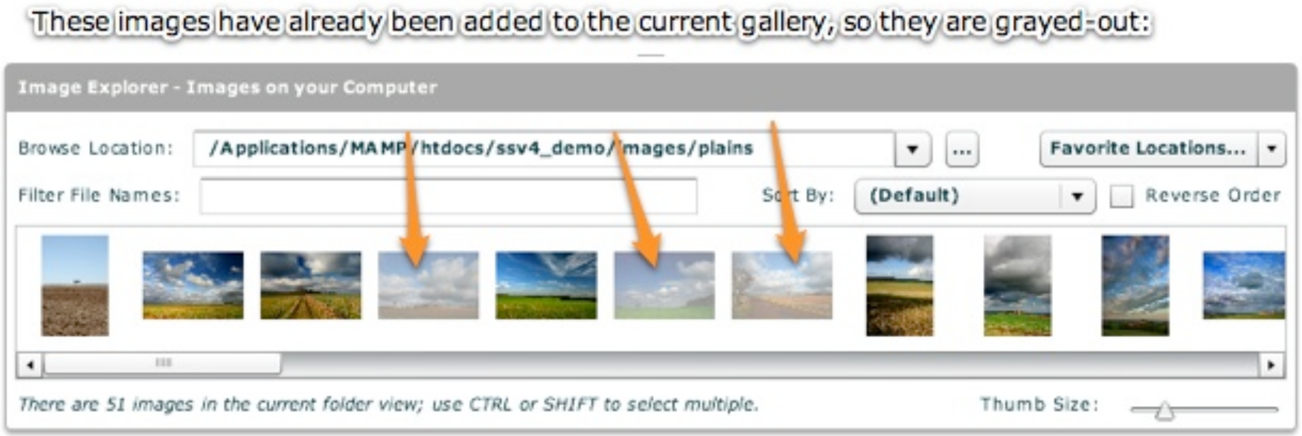
Browse Location: /Applications/MAMP/htdocs/ssv4_demo/images/plains

Filter File Names: l_52

Enter a filename filter value.

There are 3 images in the current folder view; use CTRL or SHIFT to select multiple.

If an image in the current view has already been added to the currently-selected gallery, it will be grayed-out:


These images have already been added to the current gallery, so they are grayed-out:

Browse Location: /Applications/MAMP/htdocs/ssv4_demo/images/plains

Filter File Names:

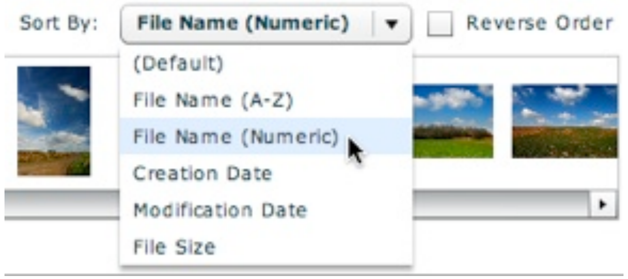
Sort By: (Default)

☐ Reverse Order

There are 51 images in the current folder view; use CTRL or SHIFT to select multiple.

Using the Pane - Sorting the Images

By default, the images are ordered based on the order in which the operating system provides the directory listing. This is usually based on the file name, but this is not always the case. By selecting a value from the Sort By menu, you can explicitly choose a sorting mode:



You can sort by file name (alphabetically or numerically), creation date, modification date, file size, or whether the image is in the current galler or not. The last option is particularly helpful when adding only new images from the directory to a gallery. Selecting the Reverse Order checkbox will reverse the order of the images in the box. **Tip:** Because the file size of images with smaller dimensions (e.g. thumbnail images) is much smaller than the file size of full-size images, you can use this to group all full-size images at one end of the box and thumbnails at the other. **Note:** The File Name (Numeric) sort option finds the first group of numbers in the file name and sorts numerically by this group. You can use this option even if the file names contain more than just numbers.

Using the Pane - Selecting Images to Use

Once you've found the location of the images you want to use, you'll need to select all of the images you want to use so you can drag them all at once. At this point, it can be helpful to expand the size of the Image Explorer Pane (using the resize handle) so that a grid of images is visible:

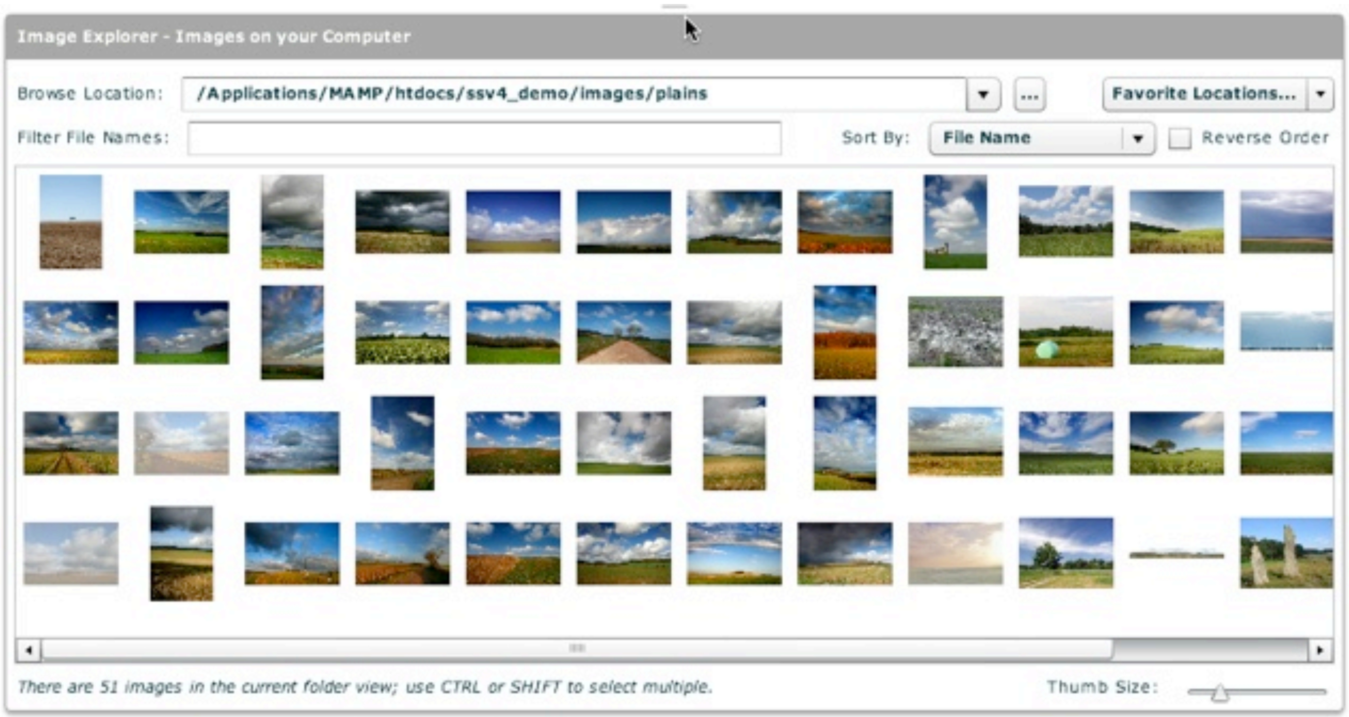


Image Explorer - Images on your Computer

Browse Location: /Applications/MAMP/htdocs/ssv4_demo/images/plains

Filter File Names:

Sort By: File Name

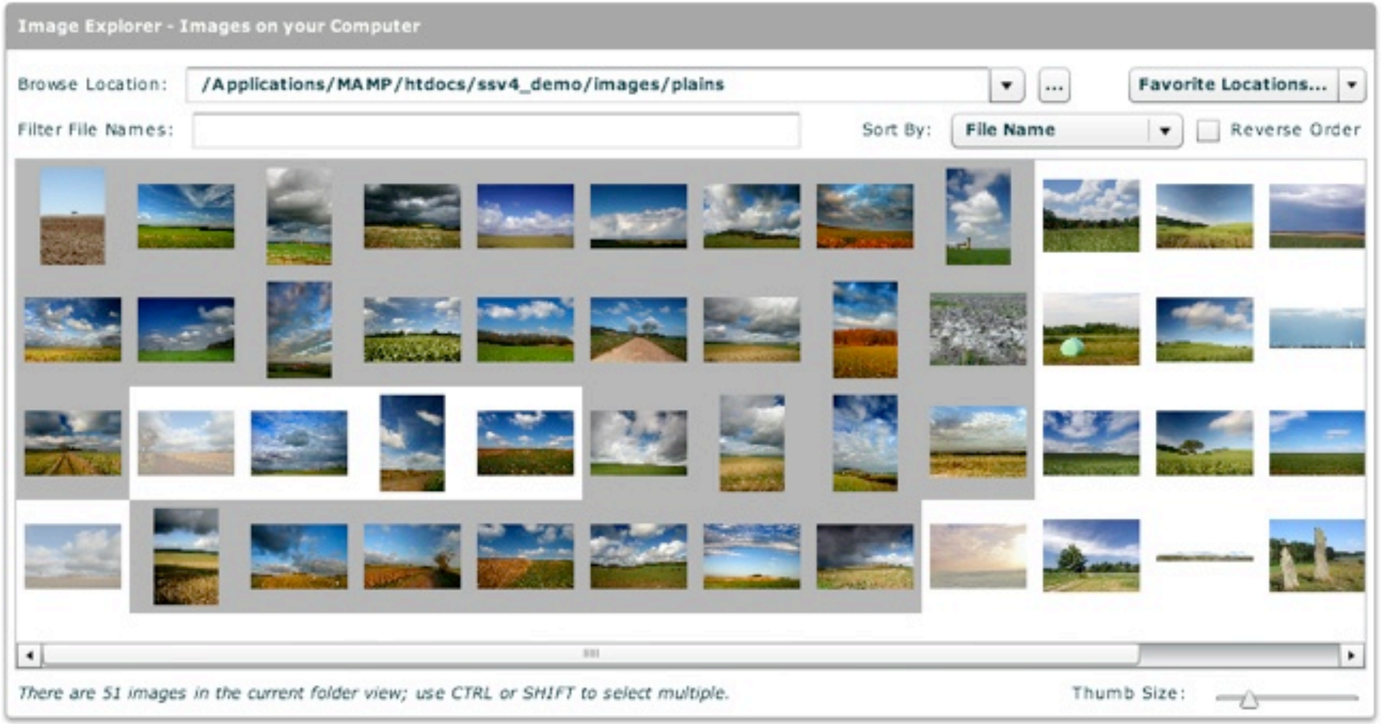
□ Reverse Order

There are 51 images in the current folder view; use CTRL or SHIFT to select multiple.

To select multiple images, you will use the CTRL (CMD on Macintosh) or SHIFT keys. The CTRL key allows you to select multiple images one-by-one, whereas the SHIFT key allows you to select all images between the current selection and the place where you click with the SHIFT key down. You can then use the CTRL key to modify the selection if you want to exclude some images within the selected block.

Tip: To use all images in a folder, click the first image, then scroll to the other end and click the final image with the SHIFT key depressed.

Shown below is an example of a large multiple selection:



There are 51 images in the current folder view; use CTRL or SHIFT to select multiple.

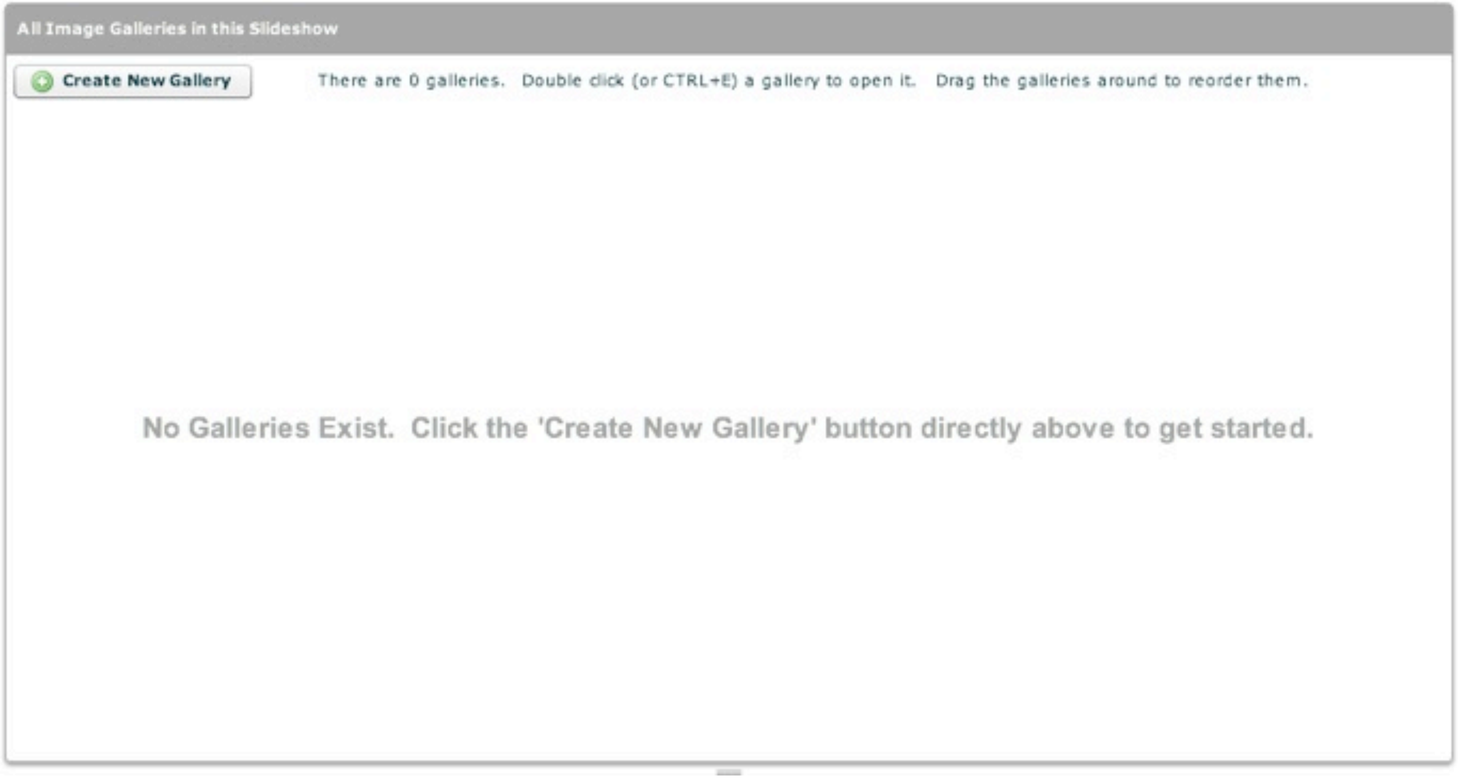
Once you have a selection, you're ready to drag it into a gallery. To learn about creating a gallery to drag the images into, proceed to the Gallery Management Pane section.

Using the External Wizard to Create and Edit Slideshows > Section-by-Section Usage Guide > Using the 'Images / Galleries' Section > **The Gallery Management Pane**

Gallery Management Pane

This topic provides a high-level overview of gallery management. Specific tasks are described in further detail in the various subtopics.

The Gallery Management Pane is where you will actually create the galleries and manage their constituent images. When you first create a slideshow, the pane appears as follows:



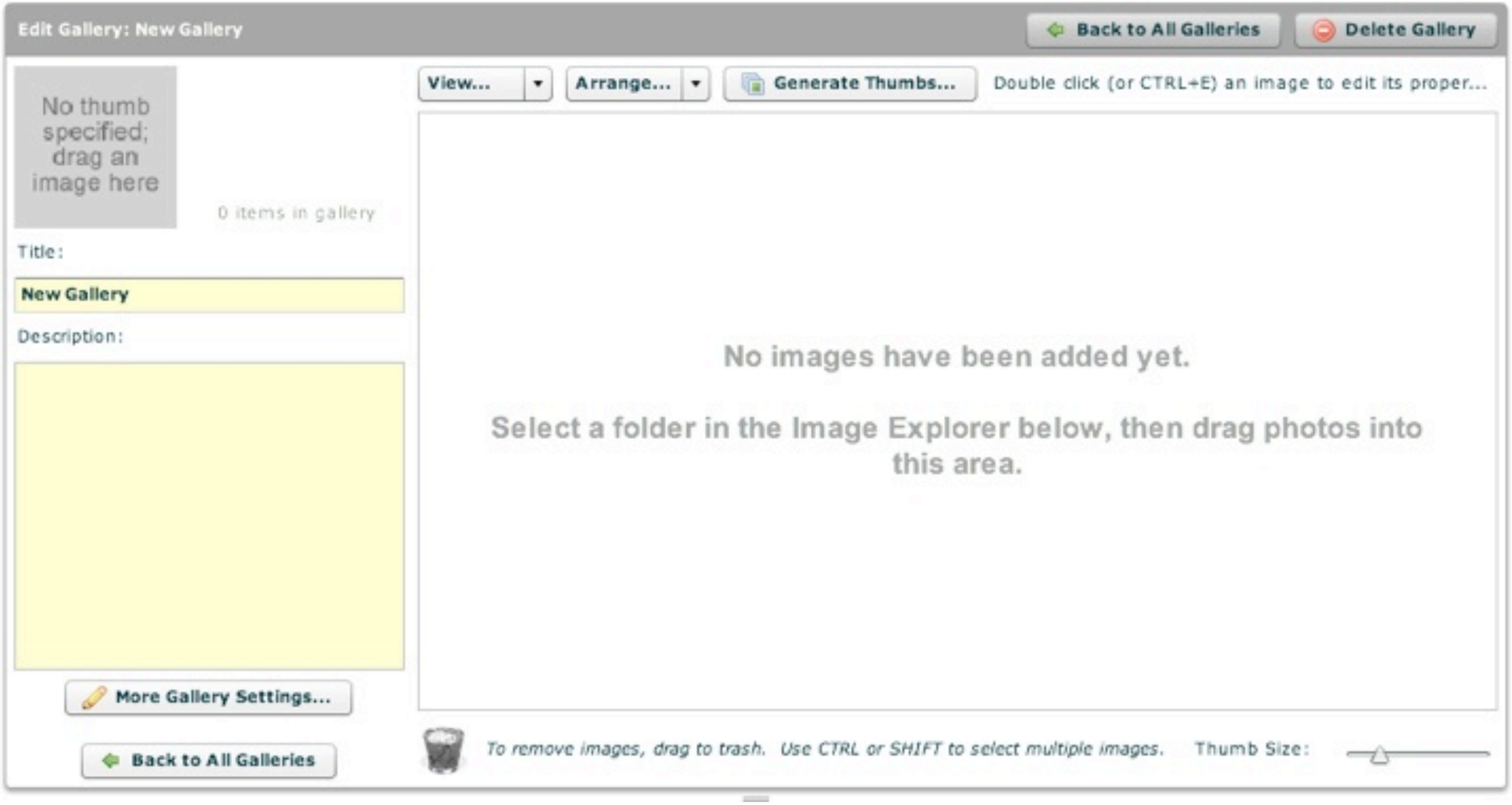
There are 0 galleries. Double click (or CTRL+E) a gallery to open it. Drag the galleries around to reorder them.

No Galleries Exist. Click the 'Create New Gallery' button directly above to get started.

Creating a New Gallery / Gallery Basics

To get started, click the Create New Gallery button:

A new gallery will be created and you'll be transferred to the single-gallery-edit view:



Double click (or CTRL+E) an image to edit its proper...

No thumb specified; drag an image here

0 items in gallery

Title:

New Gallery

Description:

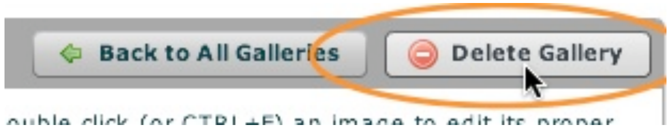
No images have been added yet.

Select a folder in the Image Explorer below, then drag photos into this area.

To remove images, drag to trash. Use CTRL or SHIFT to select multiple images.

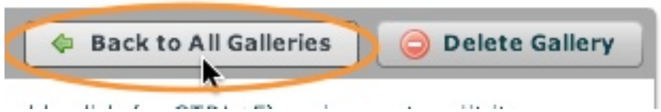
To learn more about editing the gallery's properties, see the **Editing an Individual Gallery** section below. To learn about adding images to the gallery, see the **Adding Images to a Gallery** section below.

If you wish to **remove a gallery**, you can click the Delete Gallery button in the upper-right corner:



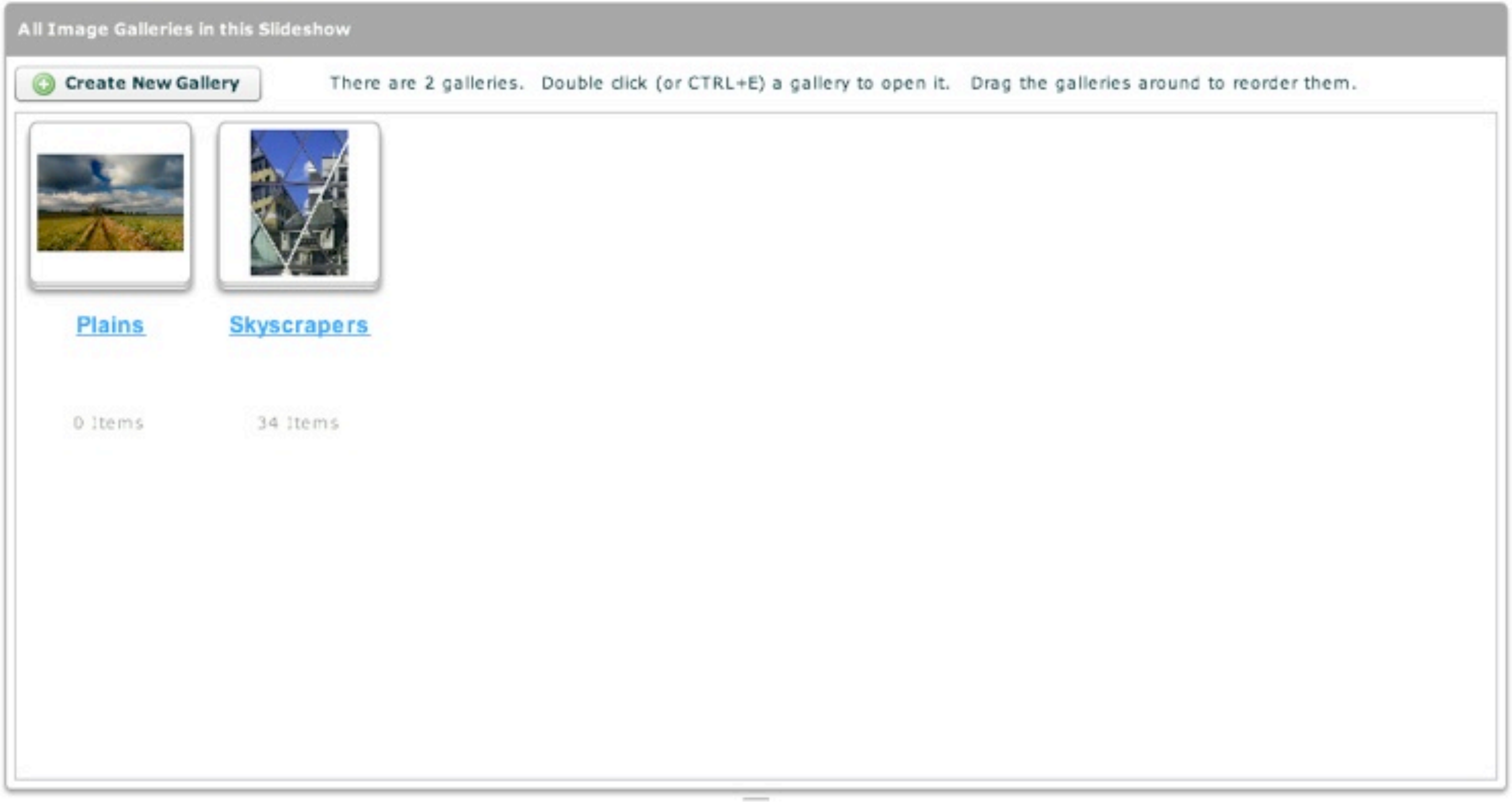
You will be promopted to confirm the deletion. Removing a gallery only removes the image references inside your slideshow; it does not modify any image files. Removing a gallery cannot be undone, so do it with caution!

To return to the All Galleries view, click the 'Back to All Galleries' button. (Note: There are actually two of these buttons, one at the upper-right and one on the bottom-left. They both have the same effect.)   To learn more about using the All Galleries view, see the **Managing All Galleries** section below.



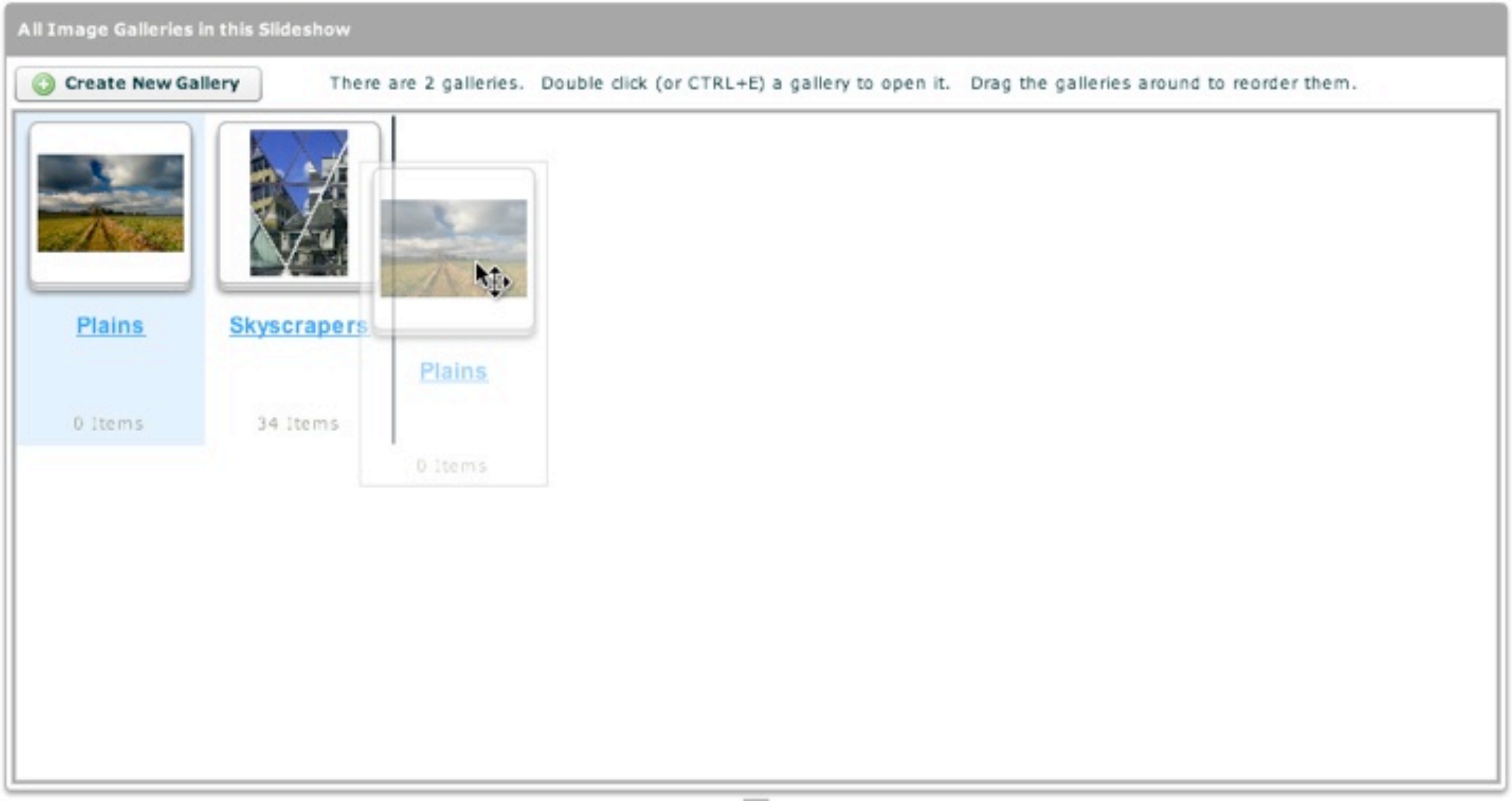
Managing All Galleries

The All Galleries view is the initial screen in the tab if you have already created galleries.   Otherwise, it is accessed by clicking the Back to All Galleries button as mentioned above.   Assuming you have added a couple of galleries, the section appears like this:



There are 2 galleries. Double click (or CTRL+E) a gallery to open it. Drag the galleries around to reorder them.

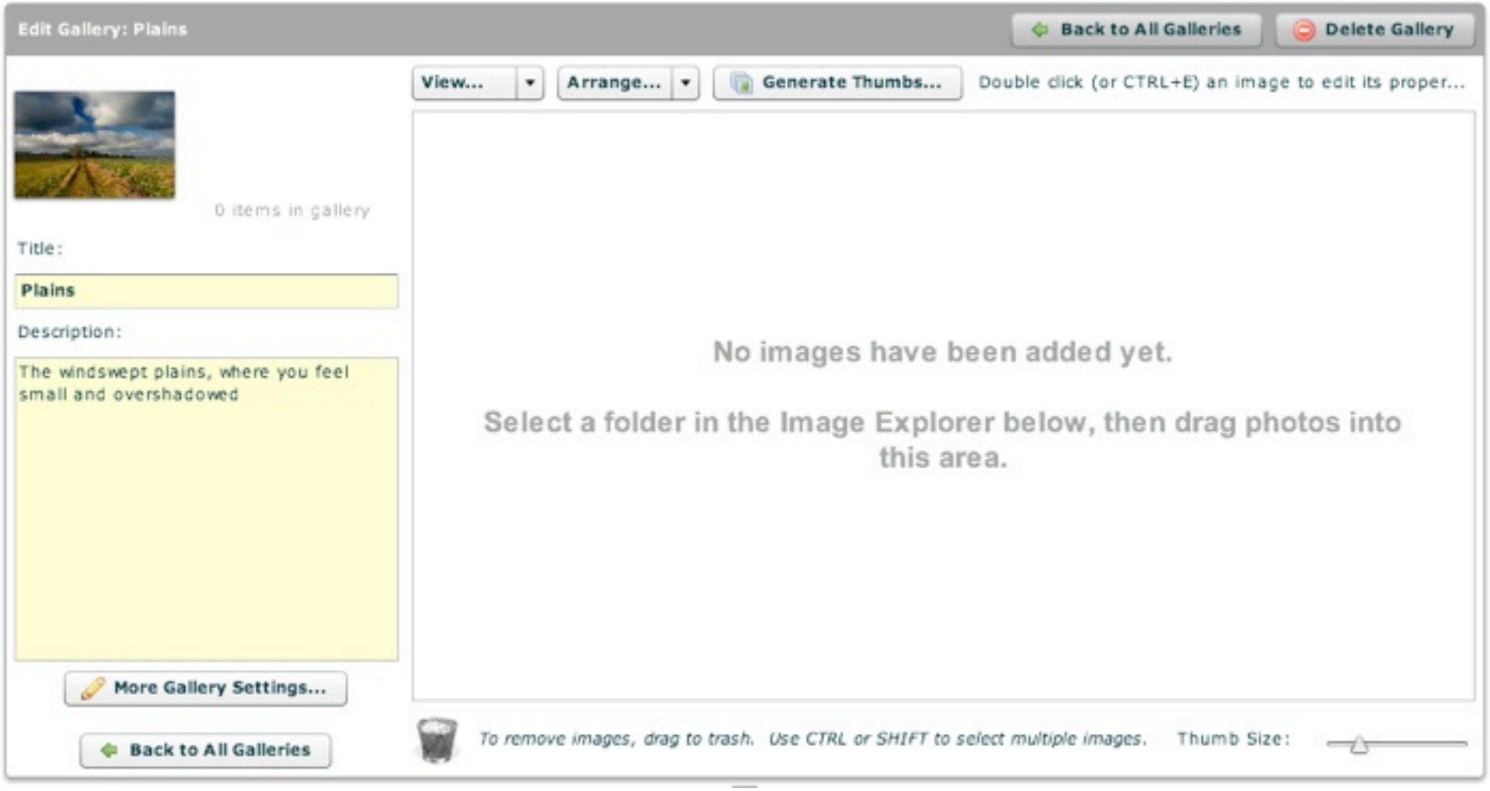
You can reorder the galleries by dragging and dropping them:



All Image Galleries in this Slideshow

There are 2 galleries. Double click (or CTRL+E) a gallery to open it. Drag the galleries around to reorder them.

To edit and individual gallery, either double-click on it or click it and hit CTRL+E (on Macintosh, CMD+E). This will take you to the single-gallery-edit view:



Double click (or CTRL+E) an image to edit its proper...

Title:

Plains

Description:

The windswept plains, where you feel small and overshadowed

No images have been added yet.

Select a folder in the Image Explorer below, then drag photos into this area.

To remove images, drag to trash. Use CTRL or SHIFT to select multiple images.

Editing an Individual Gallery's Properties

Once you've created a gallery, you'll want to set several gallery-specific properties. You can learn more about this in the Editing an Individual Gallery's Properties subtopic.

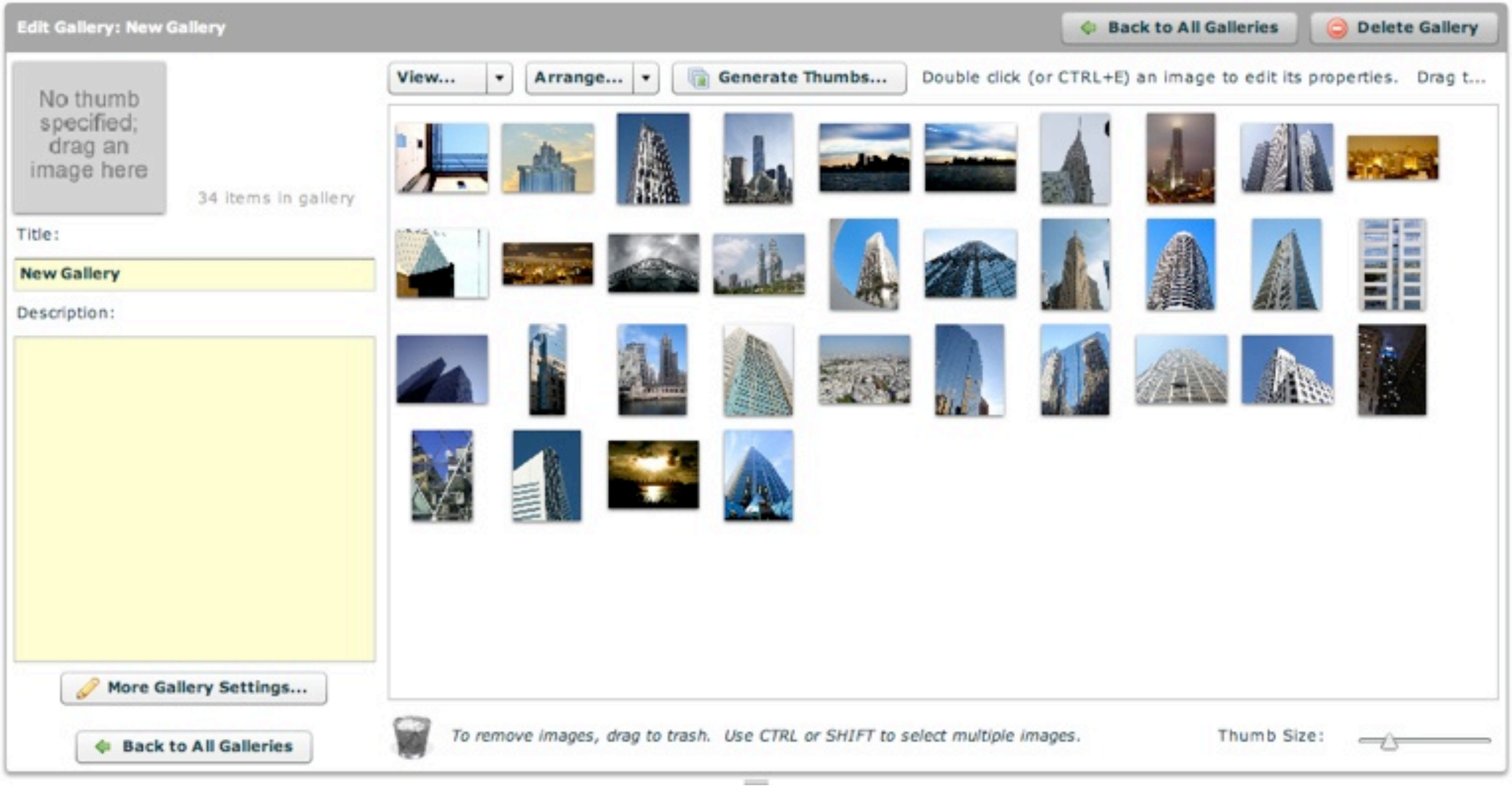
Adding Images to a Gallery

Now that you have created one or more galleries and configured them, you're ready to actually add images to the gallery. Image management is detailed in the Adding Images to a Gallery / Image Management subtopic.

Editing Individual Gallery's Properties

Selecting a gallery (by double-clicking it) brings you to the single-gallery-edit view. This is where you'll add images to the gallery and configure various gallery-specific settings. You may want to add some images first; you can learn about adding images in the next topic, Adding Images to a Gallery / Image Management.

In single-gallery-edit view, you'll see something like the screen shown below. This example shows the gallery after some images have been added:



Edit Gallery: New Gallery

No thumb specified; drag an image here

34 items in gallery

Title: New Gallery

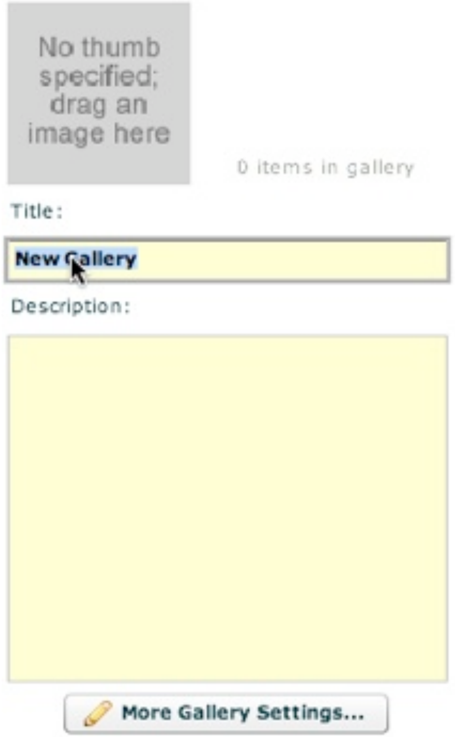
Description:

Double click (or CTRL+E) an image to edit its properties. Drag t...

To remove images, drag to trash. Use CTRL or SHIFT to select multiple images.

Setting the Gallery Title and Description

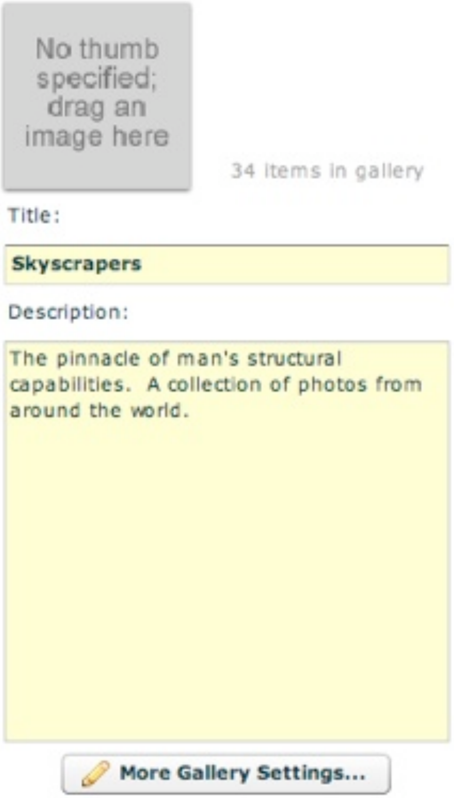
The basic gallery settings are on the left side. They include the gallery's title, description, and thumbnail image. The gallery's title and description are text values which are displayed in any gallery selection elements, describing the contents of the gallery.



Title: New Gallery

Description:

The description field is optional, but you should always enter a title value so you can identify the gallery in the All Galleries view:



No thumb specified; drag an image here

34 items in gallery

Title:

Skyscrapers

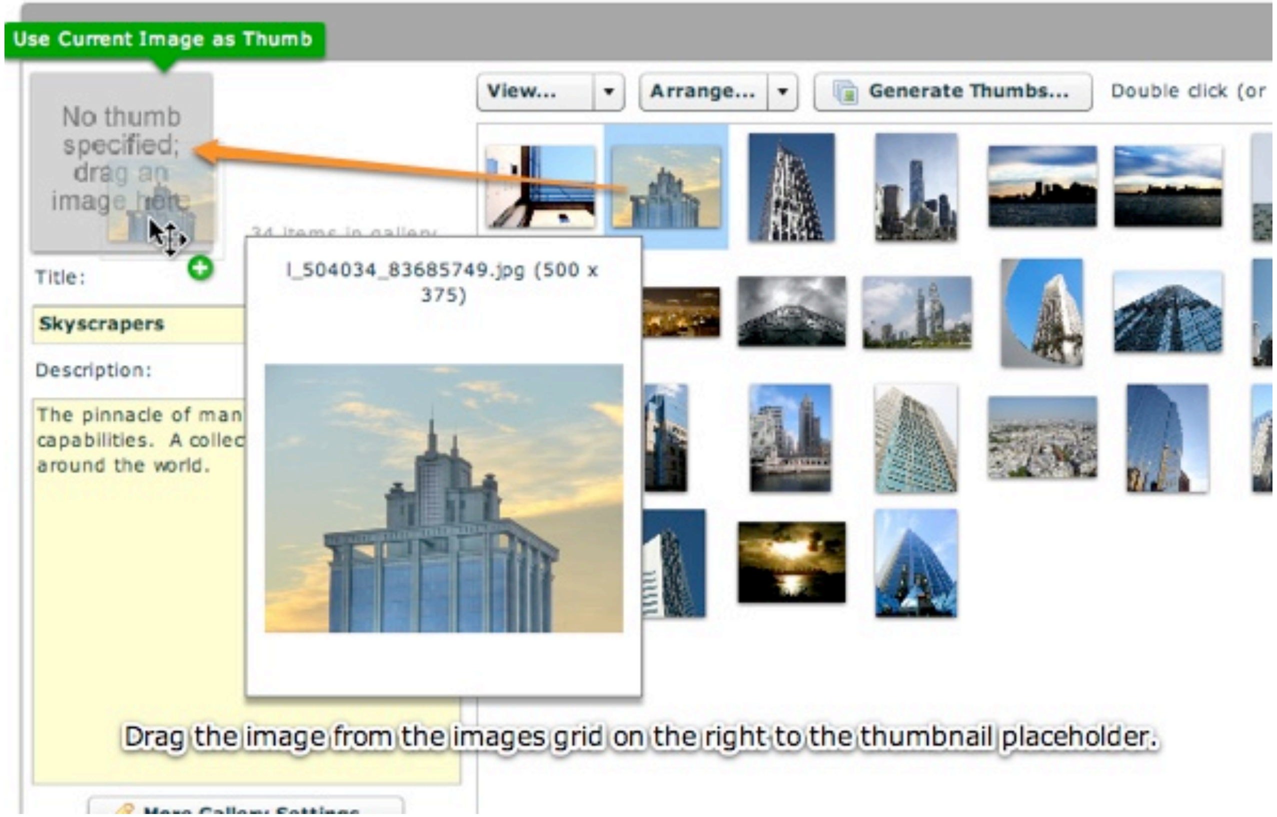
Description:

The pinnacle of man's structural capabilities. A collection of photos from around the world.

Adding a Gallery Thumbnail Image

If you have multiple galleries, you will likely want to specify a thumbnail image for the gallery. This is the thumbnail image which will be displayed with the gallery in any gallery selection elements.

If you want to use one of the gallery's images as the gallery thumbnail, you can use drag-n-drop to add it. To specify a thumbnail image by dragging and dropping, the desired image must have been already added to the gallery (i.e. it is in the image grid on the right side of the pane). It is recommended that you specify the thumbnail *after* generating thumbnails for the gallery. Otherwise, the full-size image will be used, increasing loading time and possibly causing pixelation due to scaling. If you're ready to use an image from the image grid on the right side, drag the image from the grid onto the gallery thumbnail (or thumbnail placeholder) on the left and drop it:



Title:

Skyscrapers

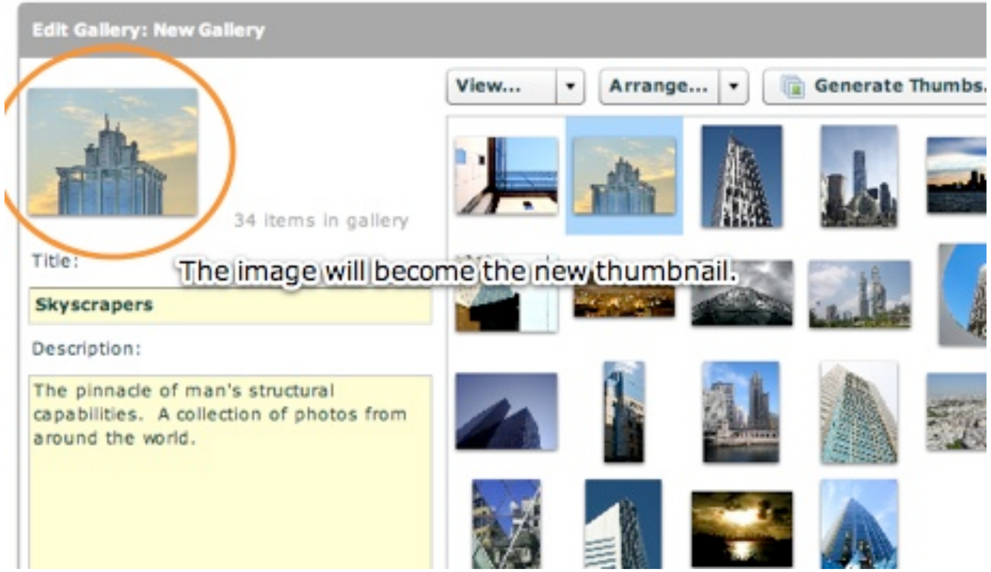
Description:

The pinnacle of man
capabilities. A collec
around the world.

l_504034_83685749.jpg (500 x 375)

Drag the image from the images grid on the right to the thumbnail placeholder.

The dropped image will become the new thumbnail image for the gallery:



34 items in gallery

The image will become the new thumbnail.

Title:

Skyscrapers

Description:

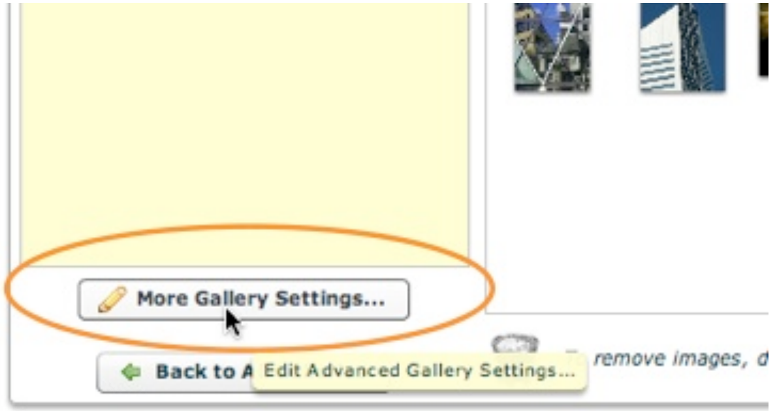
The pinnacle of man's structural capabilities. A collection of photos from around the world.

If you want to change the thumbnail, you can repeat the drag-n-drop process using any other image in the gallery.

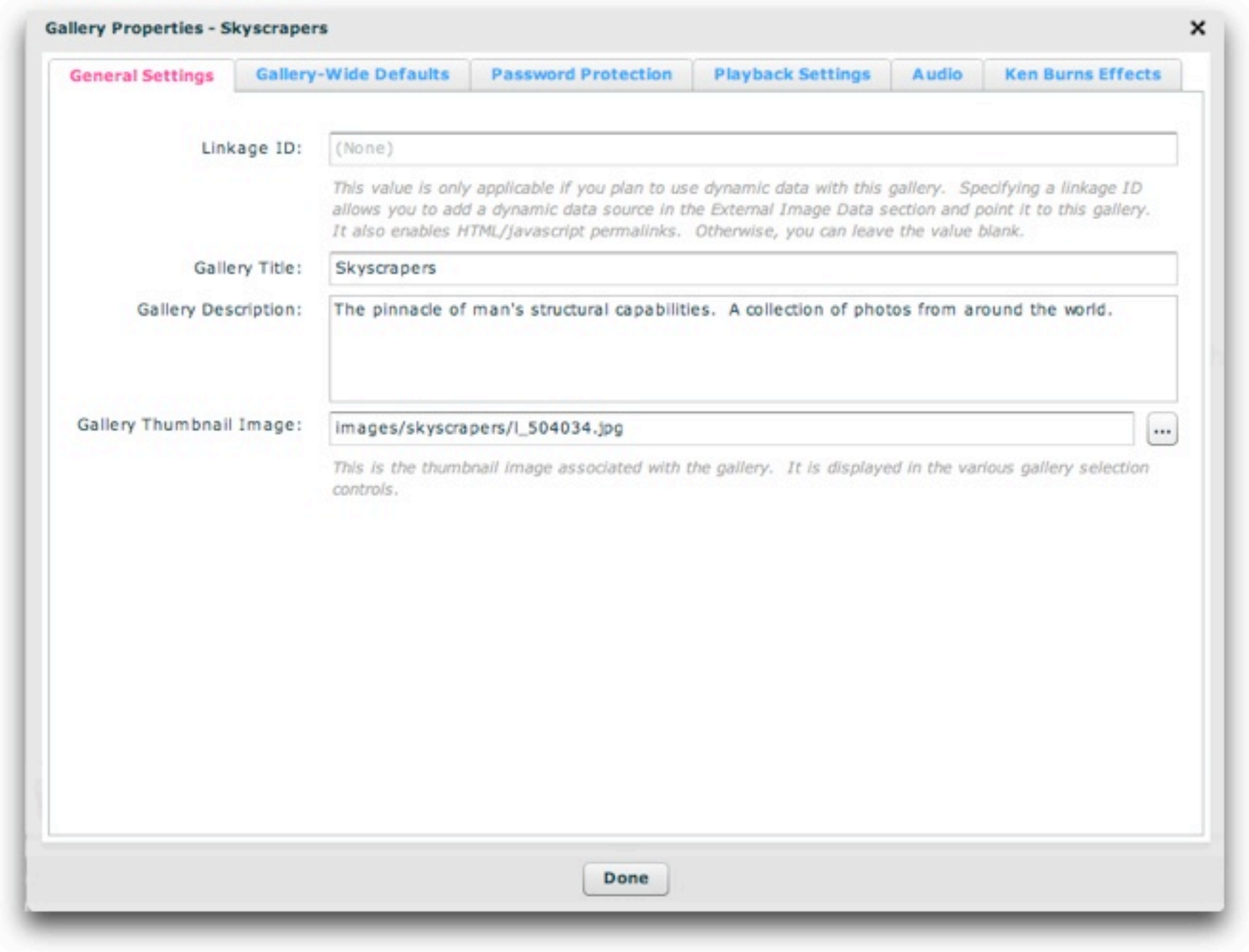
Important Note: You cannot drag images directly from the Image Explorer pane onto the drag-n-drop thumbnail area; the image must be in the images grid on the right. If you want to use an image as the gallery thumbnail which is *not* in the images grid, you can manually specify a thumbnail URL in the **More Gallery Settings** dialog (see below for information about opening and using the dialog).

Specifying Advanced Gallery Settings

There are many more advanced gallery-specific settings which can be specified for each gallery. To access these optional settings, launch the Gallery Properties dialog by clicking the **More Gallery Settings** button on the bottom left:



The Gallery Properties dialog looks like this:



Linkage ID: (None)

This value is only applicable if you plan to use dynamic data with this gallery. Specifying a linkage ID allows you to add a dynamic data source in the External Image Data section and point it to this gallery. It also enables HTML/javascript permalinks. Otherwise, you can leave the value blank.

Gallery Title: Skyscrapers

Gallery Description: The pinnacle of man's structural capabilities. A collection of photos from around the world.

Gallery Thumbnail Image: images/skyscrapers/l_504034.jpg

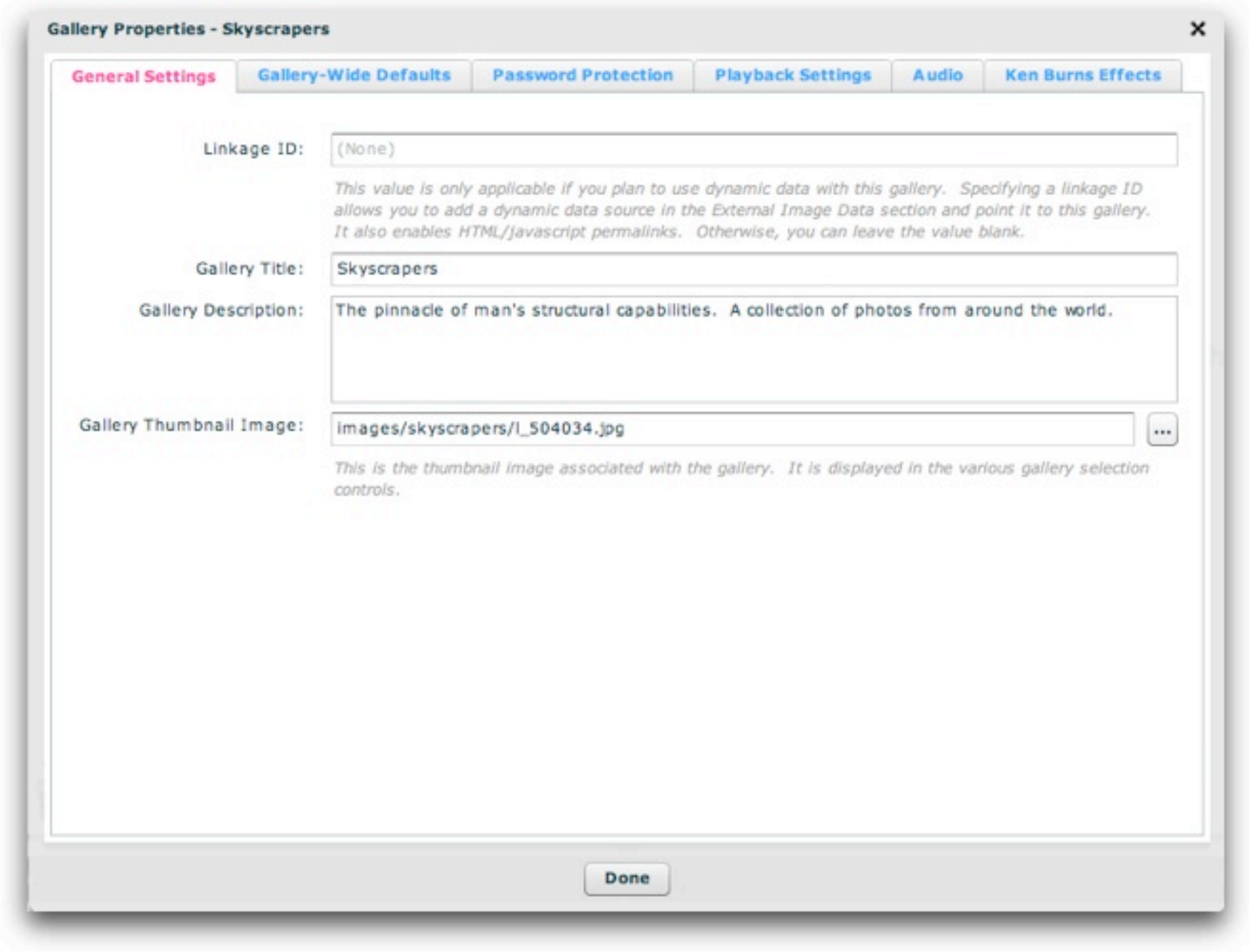
This is the thumbnail image associated with the gallery. It is displayed in the various gallery selection controls.

All of the settings in each of this dialog's tabs are covered in the subtopics of this topic; browse them for more information.

Using the External Wizard to Create and Edit Slideshows > Section-by-Section Usage Guide > Using the 'Images / Galleries' Section > The Gallery Management Pane > Editing an Individual Gallery's Properties > **Advanced Properties - General Settings**

Advanced Properties - General Settings

The General Settings tab of the Gallery Properties dialog appears as follows:



Linkage ID: (None)

This value is only applicable if you plan to use dynamic data with this gallery. Specifying a linkage ID allows you to add a dynamic data source in the External Image Data section and point it to this gallery. It also enables HTML/Javascript permalinks. Otherwise, you can leave the value blank.

Gallery Title: Skyscrapers

Gallery Description: The pinnacle of man's structural capabilities. A collection of photos from around the world.

Gallery Thumbnail Image: images/skyscrapers/l_504034.jpg

This is the thumbnail image associated with the gallery. It is displayed in the various gallery selection controls.

Each of the settings control the following:

Property	Description
Linkage ID	This **optional** value is only applicable if you plan to use dynamic data with the gallery or if you wish to reference the gallery with a permalink or through javascript interfacing. When you create and use a dynamic data source, the data can be added to a new gallery or added to an existing gallery at runtime. If it is added to a new gallery, you won't have control over the gallery settings such as title, description, thumbnail, etc. If you want control over these settings, you must create a gallery in the wizard, assign it a linkage id value, then associate this gallery linkage ID with your dynamic data source in the External Image Data section. Then, when the dynamic data is loaded, the image information will be added to the gallery you created and any explicitly-specified parameters will not be modified. If an ID is specified, the gallery can later be referenced by using this value in a permalink (e.g. linking directly to a specific gallery) or a javascript interface call. See the linked topics for more information about how these features work. Note: Only letters, numbers and underscores may be used in this value.
Gallery Title	This is the title value associated with the gallery. This value will be displayed in any gallery selection elements as the gallery's title. The value can contain HTML tags.
Gallery Description	This is the text description value associated with the gallery. This value will be displayed to provide information about the gallery in some gallery selection elements. The value can contain HTML tags.
Gallery Thumbnail Image	This is the URL to the gallery's thumbnail image. Clicking the **'...'** button to the right of the box will open a browse dialog. This image is displayed with the gallery in various gallery selection elements.

XML Equivalents (for Advanced Users and Developers)

The settings above map to the following XML attributes on each `<gallery ... > ... </gallery>` node:

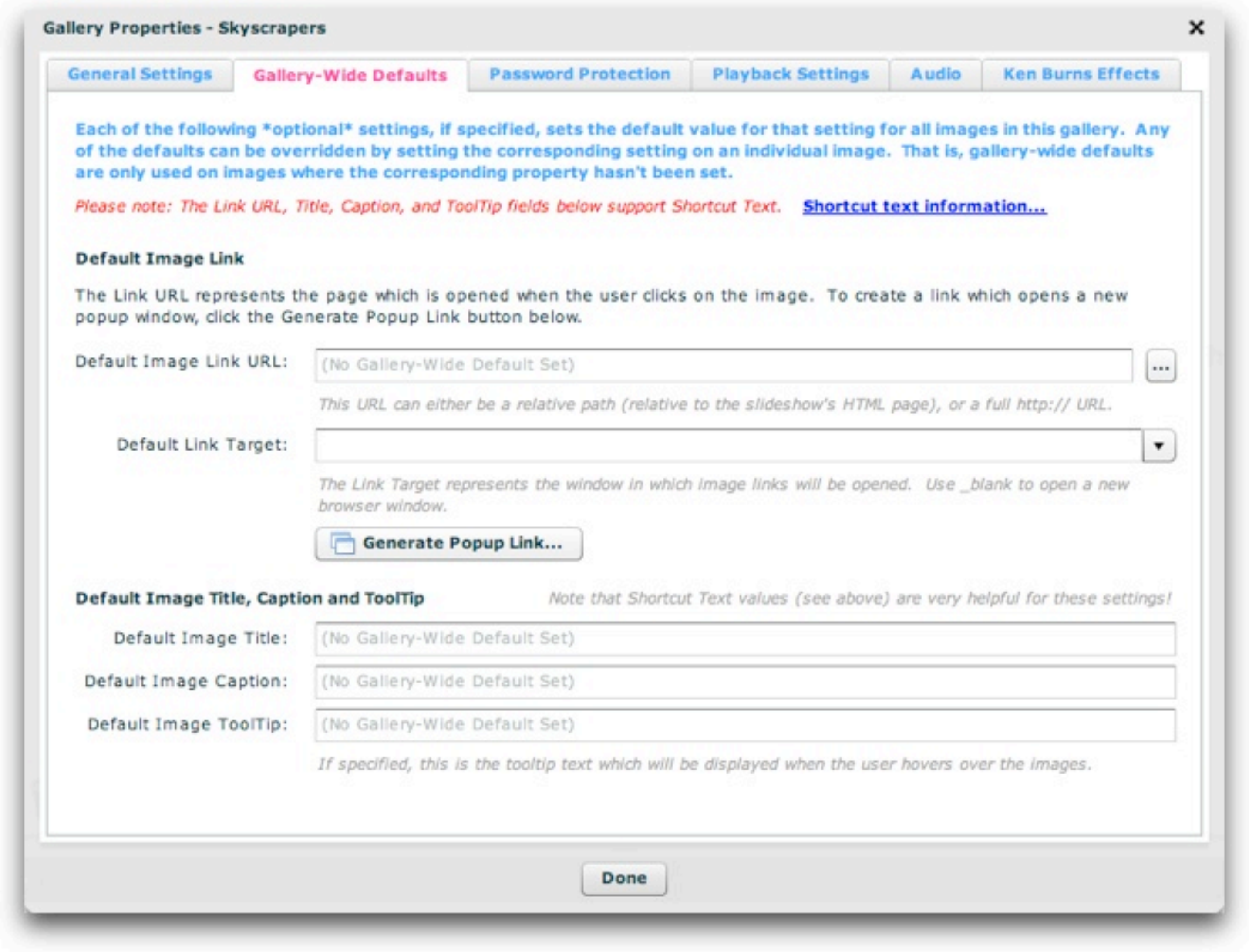
Property	Equivalent Gallery Attribute
Linkage ID	`@id`
Gallery Title	`@title`
Gallery Description	`@description`
Gallery Thumbnail Image	`@thumb`

Note that the '@' is part of customary syntactical representation, and is not part of each attribute name.

For more detailed XML information, see the Developer Reference.

Advanced Properties - Gallery-Wide Defaults

The Gallery-Wide Defaults tab of the Gallery Properties dialog appears as follows:



Gallery Properties - Skyscrapers

Each of the following *optional* settings, if specified, sets the default value for that setting for all images in this gallery. Any of the defaults can be overridden by setting the corresponding setting on an individual image. That is, gallery-wide defaults are only used on images where the corresponding property hasn't been set.

Please note: The Link URL, Title, Caption, and ToolTip fields below support Shortcut Text. Shortcut text information...

Default Image Link

The Link URL represents the page which is opened when the user clicks on the image. To create a link which opens a new popup window, click the Generate Popup Link button below.

Default Image Link URL: (No Gallery-Wide Default Set)

This URL can either be a relative path (relative to the slideshow's HTML page), or a full http:// URL.

Default Link Target:

The Link Target represents the window in which image links will be opened. Use _blank to open a new browser window.

Default Image Title, Caption and ToolTip

Note that Shortcut Text values (see above) are very helpful for these settings!

Default Image Title: (No Gallery-Wide Default Set)

Default Image Caption: (No Gallery-Wide Default Set)

Default Image ToolTip: (No Gallery-Wide Default Set)

If specified, this is the tooltip text which will be displayed when the user hovers over the images.

In this section, gallery-wide defaults can be specified for five image-specific settings. These optional values provide default values for all images in the gallery that have no image-specific value set. Defaults are applicable only for images that have no image-specific setting. Gallery-wide defaults can be overridden by setting the value on an individual image -- that is, the defaults will be ignored wherever an image-specific value is set.

When using gallery-wide defaults, it is often helpful to use replacement values. These values provide shortcuts to other image-related properties (such as metadata values). For example, this can be used to integrate metadata into the captions for all images in a gallery. For more information about available shortcut values, reference the image-specific property pages linked in the table below.

Each of the settings are detailed in the following table. Reference the documentation for the corresponding image-specific properties for more information about each setting.

Property	Description
Default Image Link URL	This optional value is the default link for all images in the gallery. It can be overridden on a per-image basis by setting the link on an individual image. The link value represents the page that is opened when an image in the gallery is clicked; to learn more about this setting and the Generate Popup tool, see the image-specific property documentation for the Image Link.
Default Link Target	The link target represents the window in which image links (links applied to images in the gallery) will be opened. This optional setting is the default value for all images in the gallery. It can be overridden on a per-image basis by setting the link target on an individual image. The default value of `_self` will open the link in the current window. Using `_blank` will open the link in a new browser window. Using `_parent` will open the link in the parent page if using frames. Specifying any other value will control a specific frame in a frameset. It is best to use gallery-wide defaults for situations where you want the same link on every image. It is much faster to set the gallery-wide default than to set the same value on every image, plus updates are much easier. **Tip:** For control over the appearance and size of the new window, you can use a javascript call in the image link to open a true popup window. For more information, see the documentation on the Image Link in the Editing Image Properties section.
Default Image Title	This optional value is the default title used for all images in the gallery that have no title specified. To learn more about this property, see the image-specific property documentation. Note that replacement values are very helpful in this field. Using a value here such as `{metadata.title}` allows for metadata-set titles to be integrated in a gallery.
Default Image Caption/Description	This optional value is the default caption/description used for all images in the gallery that have no caption/description specified. To learn more about this property, see the image-specific property documentation. Note that replacement values are very helpful in this field. Using a value here such as `{metadata.description}` allows for metadata-set descriptions to be integrated in a gallery.
Default Image ToolTip	This optional value is the default tooltip used for all images in the gallery that have no tooltip specified. Tooltips are the text displayed in a small popup when the user hovers over an image. To learn more about this property, see the image-specific property

	documentation. Note that replacement values are very helpful in this field. This can also be used to provide a standard message when a user hovers over an image. It is much faster to set the gallery-wide default than to set the same value on every image, plus updates are much easier.

XML Equivalents (for Advanced Users and Developers)

The settings above map to the following XML attributes on each `<gallery ... > ... </gallery>` node:

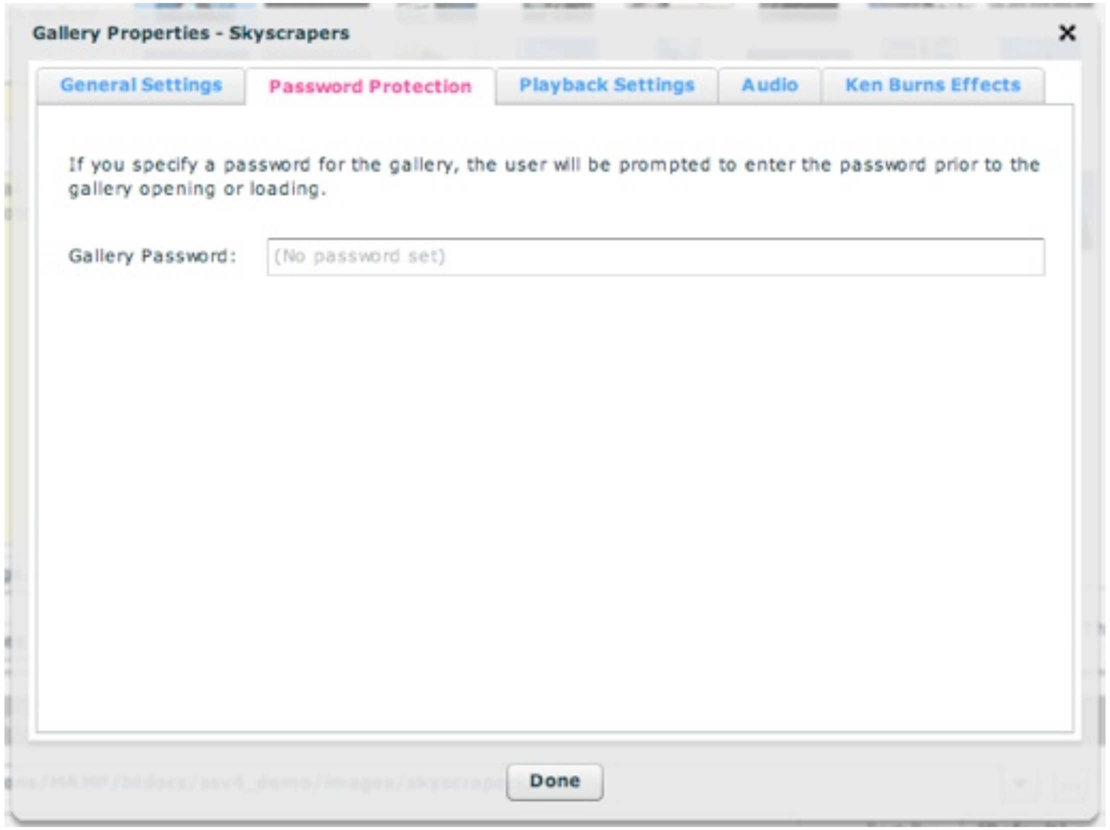
Property	Equivalent Gallery Attribute
Default Image Link URL	`@defaultLink`
Default Link Target	`@linkTarget`
Default Image Title	`@defaultTitle`
Default Image Caption/Description	`@defaultCaption`
Default Image ToolTip	`@defaultToolTip`

Note that the '@' is part of customary syntactical representation, and is not part of each attribute name.

For more detailed XML information, see the Developer Reference.

Advanced Properties - Password Protection

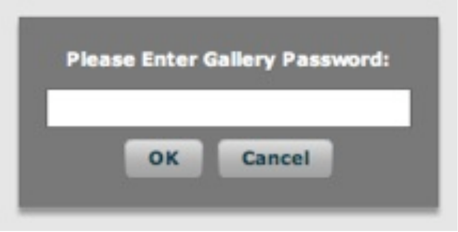
The Password Protection tab of the Gallery Properties dialog appears as follows:



If you specify a password for the gallery, the user will be prompted to enter the password prior to the gallery opening or loading.

Gallery Password: (No password set)

This is where you can optionally specify a gallery password. If a gallery password has been specified, the user will receive a password prompt when attempting to open the gallery, and will be required to enter the password correctly:



Note: Passwords are encrypted.

XML Equivalents (for Advanced Users and Developers)

The settings above map to the following XML attributes on each `<gallery ... > ... </gallery>` node:

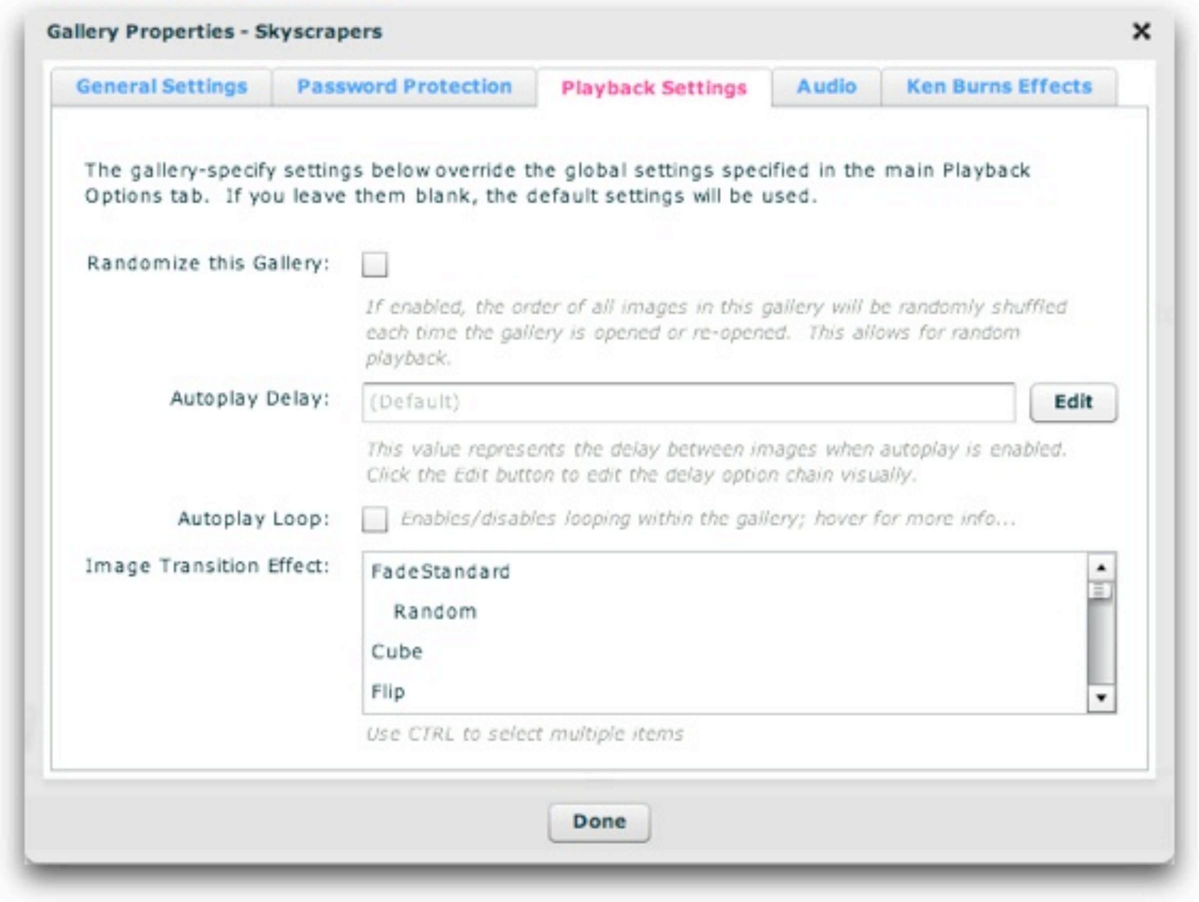
Property	Equivalent Gallery Attribute
Gallery Password	`@password`

Note that the '@' is part of customary syntactical representation, and is not part of each attribute name.

For more detailed XML information, see the Developer Reference.

Advanced Properties - Playback Settings

The Playback Settings tab of the Gallery Properties dialog appears as follows:



The gallery-specify settings below override the global settings specified in the main Playback Options tab. If you leave them blank, the default settings will be used.

Randomize this Gallery: □

If enabled, the order of all images in this gallery will be randomly shuffled each time the gallery is opened or re-opened. This allows for random playback.

Autoplay Delay: (Default)

This value represents the delay between images when autoplay is enabled. Click the Edit button to edit the delay option chain visually.

Autoplay Loop: □ *Enables/disables looping within the gallery; hover for more info...*

Image Transition Effect: FadeStandard, Random, Cube, Flip

Use CTRL to select multiple items

These settings allow you to override the global playback settings (as specified in the main 'Global Configuration & Playback Settings' tab) on a per-gallery basis. For more detailed information about these settings, see the **Using the 'Global Configuration and Playback Settings' Section** area of the manual.

Each of the settings controls the following:

Property	Description
Randomize this Gallery	If selected, the gallery is randomized each time it is opened. In other words, the order of all slides/images in the gallery is shuffled. This ensures true runtime randomization of the gallery every time it is viewed.
Autoplay Delay	When autoplay is enabled, this value specifies the delay between each image/slide. These values are specially constructed strings; clicking the Edit button allows you to visually edit the value. To learn more, see Using the Autoplay Delay Editor.
Autoplay Loop	Enables/disables looping within the gallery. If enabled, autoplay will jump back to the first image in the gallery when the final image is reached. If disabled, autoplay will jump to the first image in the next gallery when the final image is reached.
Image Transition Effect	This parameter only applies if you're using the ImagesStandard image display element (most slideshows do). In these cases, you can specify the image transition effect for this gallery, overriding the global setting. To view a description of each effect, you can hover over the effect name. Or, to view a summary of all effects, see the Setting the Image Transition topic in the *Using the 'Global Configuration and Playback Settings' Section* topic. If you don't wish to have the description tooltips, you can temporarily disable the tooltips by selecting File > Enable/Disable All Tooltips on the program menu. Selecting multiple items (by holding down the CTRL or CMD key) will make the slideshow randomly cycle through those selected transitions. To cycle through *all* of the transitions randomly, select only the **Random** option.

XML Equivalents (for Advanced Users and Developers)

The settings above map to the following XML attributes on each `<gallery ... > ... </gallery>` node:

Property	Equivalent Gallery Attribute	Notes
Randomize this Gallery	`@randomize`	Values of `true` or `false` are supported.
Autoplay Delay	`@autoplayDelay`	To learn more about the `autoplayDelay` format, see the `com.dwuser.ss4.managers.AutoplayManager` class in the API Reference.

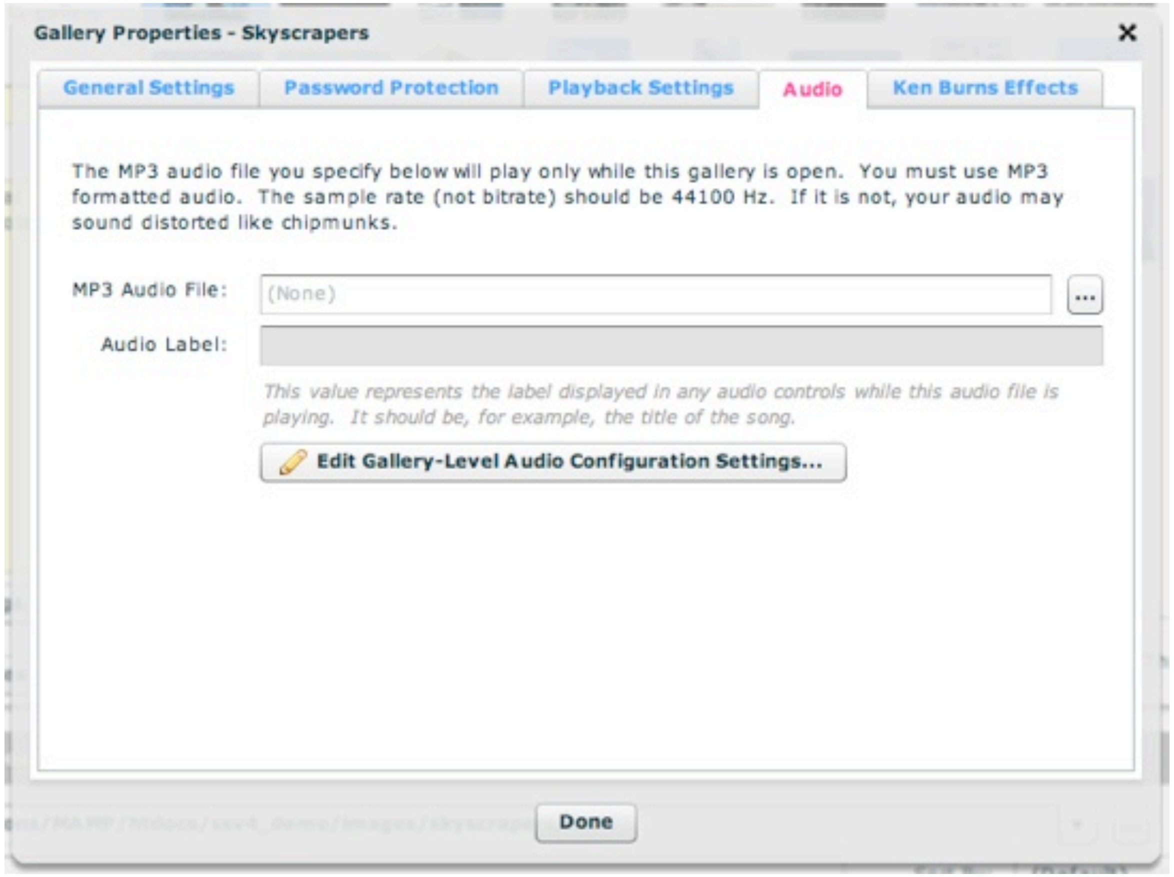
Autoplay Loop	`@autoplayLoop`	Values of `true` or `false` are supported.
Image Transition Effect	`@effectType`	If using more than one effect, each of their names must be separated with a comma (no spaces). For more information about supported transitions, see Setting the Image Transition. For example: `...effectType="FadeStandard,Cube" ...`

Note that the '@' is part of customary syntactical representation, and is not part of each attribute name.

For more detailed XML information, see the Developer Reference.

Advanced Properties - Audio

The Playback Settings tab of the Gallery Properties dialog appears as follows:



Gallery Properties - Skyscrapers

The MP3 audio file you specify below will play only while this gallery is open. You must use MP3 formatted audio. The sample rate (not bitrate) should be 44100 Hz. If it is not, your audio may sound distorted like chipmunks.

MP3 Audio File: (None)

Audio Label:

This value represents the label displayed in any audio controls while this audio file is playing. It should be, for example, the title of the song.

This is where you can specify gallery-level audio. This is the audio which plays only while this gallery is open. The audio file must be in MP3 format; for necessary formatting guidelines, see Avoiding "Chipmunks" Audio: MP3 Formatting Guidelines. Enter the URL in the box or use the browse button to the right ('...') to select a file. The URL must be relative to the HTML page which holds the slideshow.

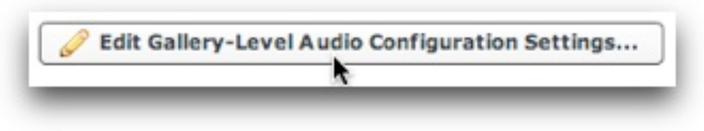
Note: If you are loading audio from a **different domain** than the one on which the slideshow is displayed, you will be affected by Flash Player crossdomain loading security restrictions. To learn more, see this topic.

Note: If you're audio plays back at a **distorted speed**, like "**chipmunks**", you need to review the MP3 formatting guidelines referenced above.

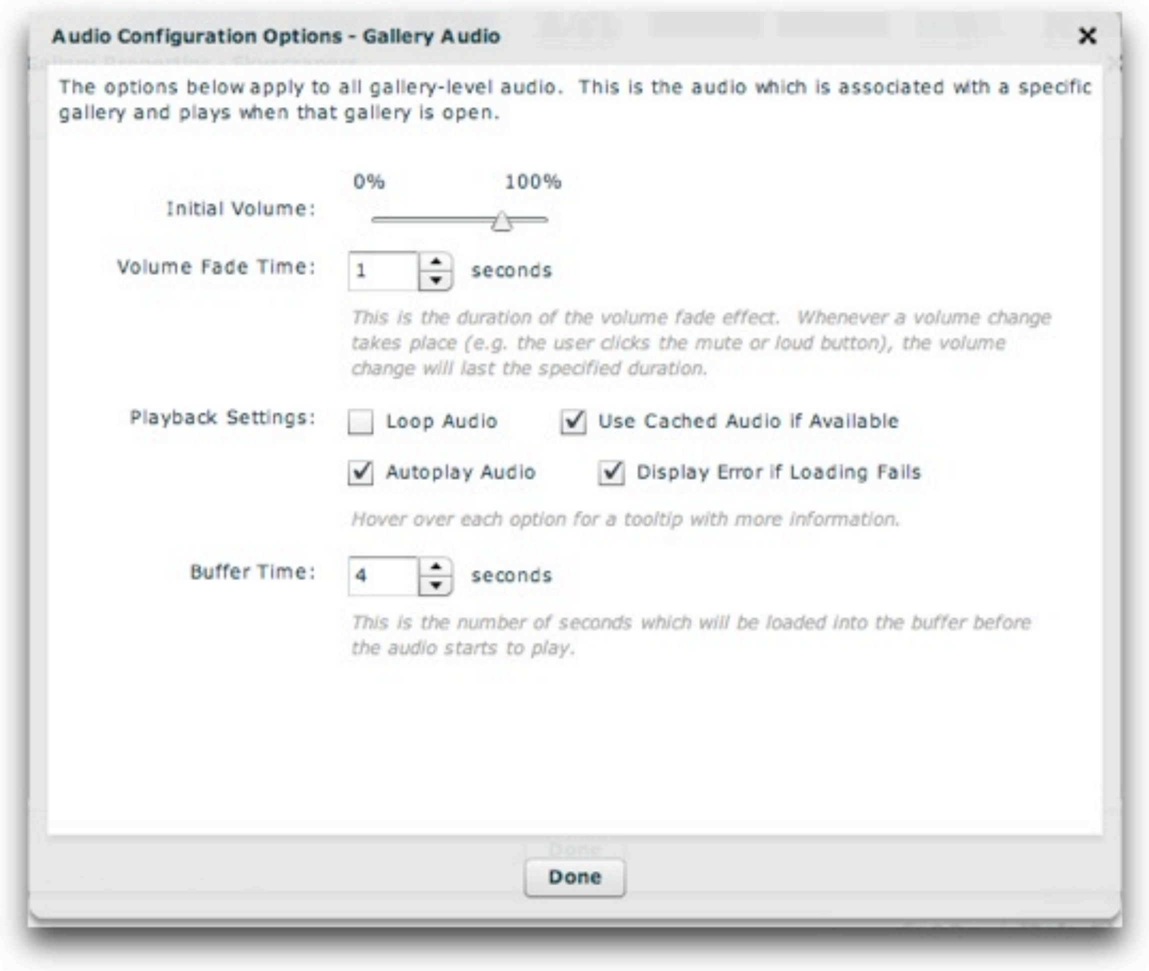
Once you specify an audio file, you can enter a string value in the Audio Label field. This value represents the label associated with the audio file; it is shown in any related audio controls while the file is playing. It should be the title of the song or the narration, for example.

Editing the Gallery-Level Audio Configuration Settings

There are several configuration settings, specific to gallery-level audio, which can be specified. These options are accessed through the 'Edit Gallery-Level Audio Configuration Settings...' button.



The dialog appears as follows. Note that the settings in this dialog apply to all gallery-level audio, not just the audio applied to a single gallery.



Audio Configuration Options - Gallery Audio

The options below apply to all gallery-level audio. This is the audio which is associated with a specific gallery and plays when that gallery is open.

Initial Volume: 0% 100%

Volume Fade Time: 1 seconds

This is the duration of the volume fade effect. Whenever a volume change takes place (e.g. the user clicks the mute or loud button), the volume change will last the specified duration.

Playback Settings: □ Loop Audio ☑ Use Cached Audio if Available

☑ Autoplay Audio ☑ Display Error if Loading Fails

Hover over each option for a tooltip with more information.

Buffer Time: 4 seconds

This is the number of seconds which will be loaded into the buffer before the audio starts to play.

Each of the options controls the following (when working in the wizard, you can also learn more about some settings directly by hovering over their names):

Property	Description
Initial Volume	This is the initial volume of the audio file when it begins playing, excepting any user adjustments via a volume control. If you have provided user controls for volume, then this is simply the starting point. A value of 0% is entirely silent, whereas 100% is full volume.
Volume Fade Time	The volume fade effect fades the volume from its current value to the new value over the duration specified by this setting. Especially if you are using mute/loud volume controls (which jump directly to a certain volume), it is recommended that you utilize this effect. This provides a polished, less jarring transition when muting and unmuting audio.

Loop Audio	Specifies if the audio should loop back to the beginning when it finishes playing.
Use Cached Audio if Available	Rarely should this setting be disabled. If enabled, the slideshow will use a cached version of the audio file from the user's internet cache when available. This will improve loading performance if the user listens to the same file more than once, making it start almost instantly on subsequent loads. This setting should only be disabled in cases where the audio file is frequently updated.
Autoplay Audio	Specifies if the audio should start playing automatically. If this option is disabled, you must give the user audio controls and they must manually play the audio.
Display Error if Loading Fails	By default, an error will be shown when the loading of the audio fails. In a production setup, you may want to disable this setting. In rare instances on certain configurations, the audio may play correcly yet an error message appears. In these cases, the problem can be avoided by disabling the error reporting.
Buffer Time	This is the number of seconds of audio which must be loaded into the buffer before the audio starts to play. Additionally, if the buffer runs out, this duration will be loaded before playback resumes. Generally the default value of about 4 seconds is a good setting; it will ensure that even on slow connections the audio doesn't cut in and out.

XML Equivalents (for Advanced Users and Developers)

The settings above map to the following XML attributes on each `<gallery ... > ... </gallery>` node:

Property	Equivalent Gallery Attribute
MP3 Audio File	`@audio`
Audio Label	`@audioLabel`

Note that the '@' is part of customary syntactical representation, and is not part of each attribute name.

The Gallery-Level Audio Configuration Settings map to the `<AudioGallery ... />` styles node within the `<styles>` section of the XML configuration file. The settings in the above table map to the following attributes on that node:

Property	Equivalent <AudioGallery ... /> Attribute	Notes
Initial Volume	`@volume`	Supported values range from 0 to 100.
Volume Fade Time	`@volumeChangeTime`	Denominated in milliseconds (whereas wizard is seconds).

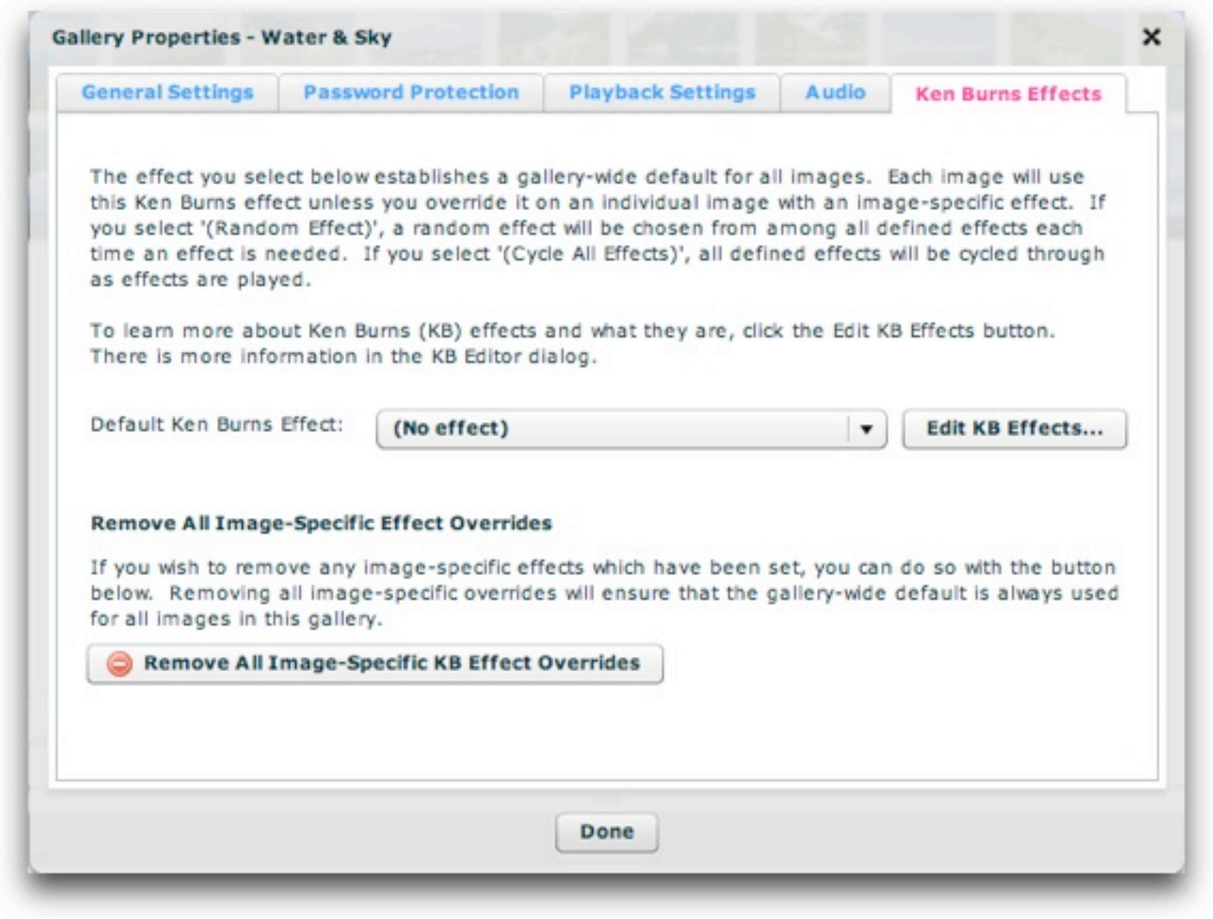
Loop Audio	`@loop`	Supported values are `true` and `false`.
Use Cached Audio if Available	`@useCache`	Supported values are `true` and `false`.
Autoplay Audio	`@autoplay`	Supported values are `true` and `false`.
Display Error if Loading Fails	`@errorReporting`	Supported values are `true` and `false`.
Buffer Time	`@buffer`	Denominated in seconds (same as wizard).

Note that the '@' is part of customary syntactical representation, and is not part of each attribute name.

For more detailed XML information, see the Developer Reference.

Advanced Properties - Ken Burns Effects

The Ken Burns Effects tab of the Gallery Properties dialog appears as follows:



The effect you select below establishes a gallery-wide default for all images. Each image will use this Ken Burns effect unless you override it on an individual image with an image-specific effect. If you select '(Random Effect)', a random effect will be chosen from among all defined effects each time an effect is needed. If you select '(Cycle All Effects)', all defined effects will be cycled through as effects are played.

To learn more about Ken Burns (KB) effects and what they are, click the Edit KB Effects button. There is more information in the KB Editor dialog.

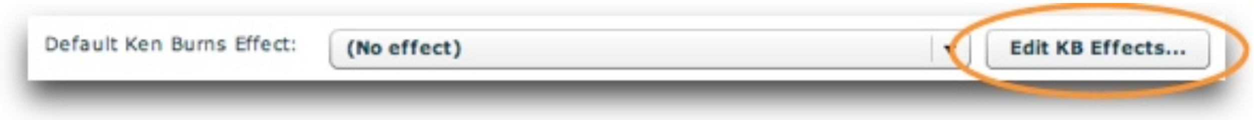
Default Ken Burns Effect: (No effect) ▾

Remove All Image-Specific Effect Overrides

If you wish to remove any image-specific effects which have been set, you can do so with the button below. Removing all image-specific overrides will ensure that the gallery-wide default is always used for all images in this gallery.

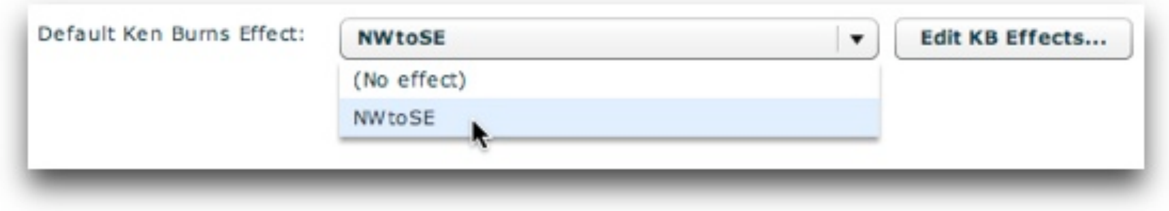
This is where you can specify a default Ken Burns effect for all images in the gallery. Ken Burns effects (KB effects) are packages which incorporate both panning and zooming. You can learn more about KB effects in the topic on using the KB Effect Editor (there is a link to this topic below). If you specify an effect in this panel, it will be used by each image in the gallery unless you override it on an individual image with an image-specific effect.

To be able to select an effect, you must first create (define) effects using the KB Effect Editor. This dialog is accessed by clicking the 'Edit KB Effects...' button:



Default Ken Burns Effect: (No effect)

This will launch the KB Effect Editor. To learn about how to use this window to define effects, see Using The Ken Burns Effect (Panning / Zooming) Editor. After creating an effect and returning to the Gallery Properties dialog, you'll be able to select the effect you just created from the dropdown list:



Default Ken Burns Effect: NWtoSE

After defining multiple KB effects, you can also choose the **(Random Effect)** option from the dropdown (this option is not shown in the above screenshot). When this option is selected, an effect will randomly be selected from among all defined effects each time an effect is needed. Or, you can also choose the **(Cycle All Effects)** option from the dropdown (this option is not shown in the above screenshot). When this option is selected, all created effects will be cycled through as KB effects are displayed.

Removing Image-Specific Effect Overrides

The 'Remove All Image-Specific KB Effect Overrides' button allows you to remove any KB effects which have been set on a per-image basis. This ensures that the gallery-wide default established above is always used for all images in the gallery.

XML Equivalents (for Advanced Users and Developers)

The settings above map to the following XML attributes on each `<gallery ... > ... </gallery>` node:

Property	Equivalent Gallery	Notes

	Attribute	
Default Ken Burns Effect	`@kbEffect`	In the XML, you can specify multiple KB effect ID values; when doing this, separate each ID with a comma (no spaces). One of these multiple effects will be randomly chosen each time an effect is to be applied. Specifying `Random` will cause a random effect to be selected from among all defined effects each time an effect is needed. Specifying `Cycle` will cause all defined effects to be cycled through as effects are needed.

Note that the '@' is part of customary syntactical representation, and is not part of each attribute name.

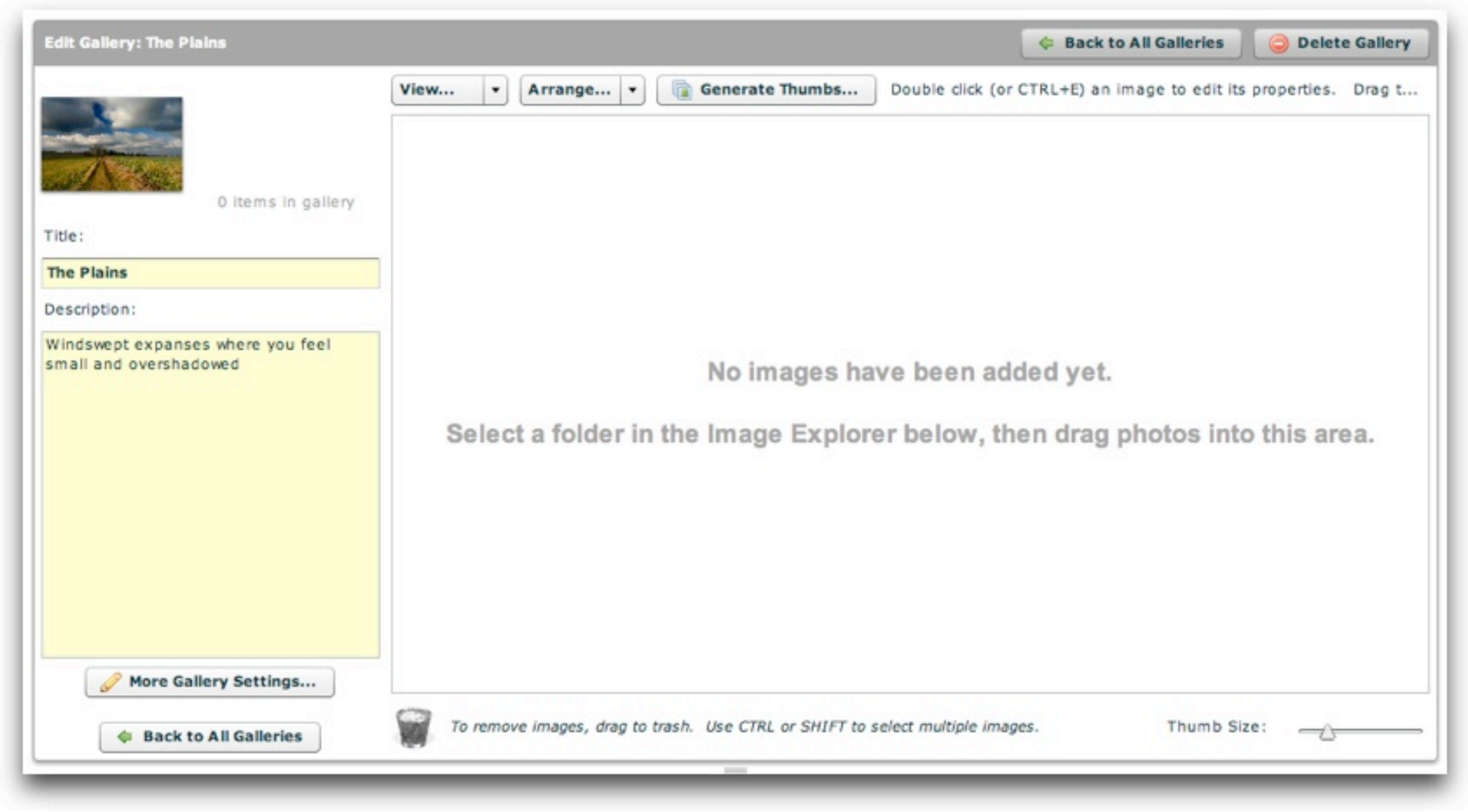
For more detailed XML information, see the Developer Reference.

Adding Images to a Gallery / Image Management

This topic covers the process of adding photos to your gallery, customizing their setup within the gallery, then managing the options for each photo. Automatic thumbnail generation is also covered.

Adding Images to a Gallery

Now for the moment you've been waiting for -- you have a gallery, and you're ready to add some images to it. You should have double-clicked on the gallery and be in single-gallery-edit view, like this:



Double click (or CTRL+E) an image to edit its properties. Drag t...

0 items in gallery

Title: The Plains

Description: Windswept expanses where you feel small and overshadowed

No images have been added yet.

Select a folder in the Image Explorer below, then drag photos into this area.

To remove images, drag to trash. Use CTRL or SHIFT to select multiple images.

As you can see, there are not yet any images in the Images Grid.

In the Image Explorer pane at the bottom, you should have already selected some images to add to the gallery. (This process was covered in detail in the Using the Image Explorer Pane topic. The folder **must be within your Dreamweaver site**!) For this example, 23 various photos have been selected using a combination of SHIFT and CTRL (CMD). These photos are ready to be added to the gallery:

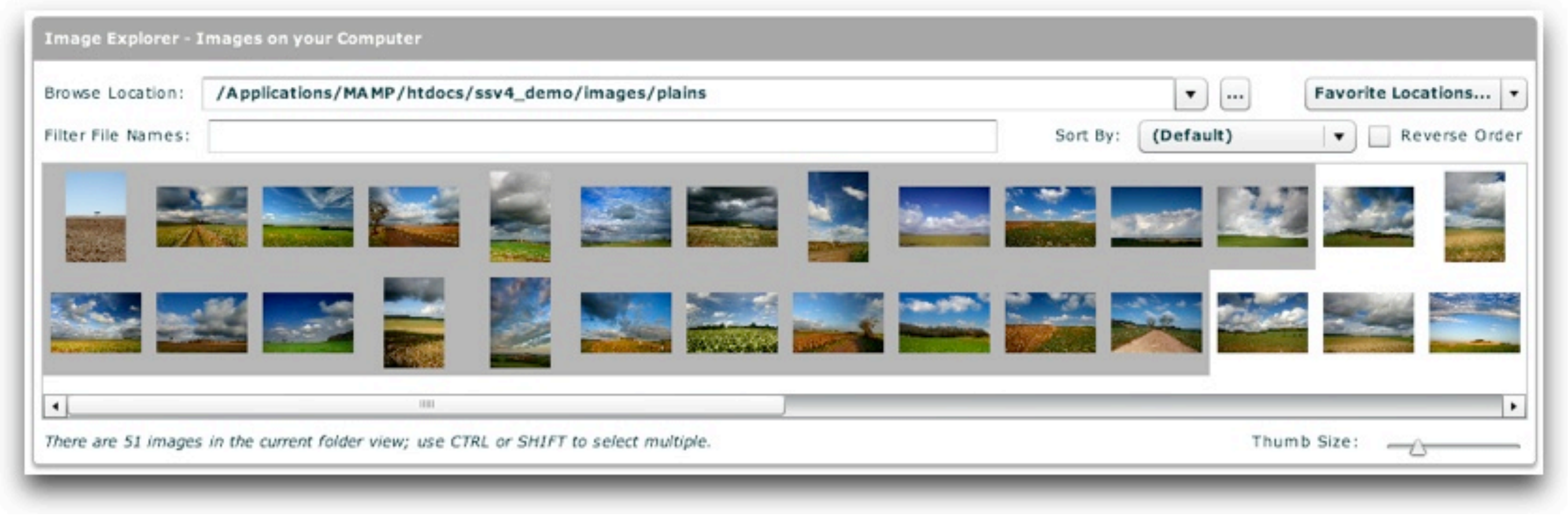


Image Explorer - Images on your Computer

Browse Location: /Applications/MAMP/htdocs/ssv4_demo/images/plains

Filter File Names:

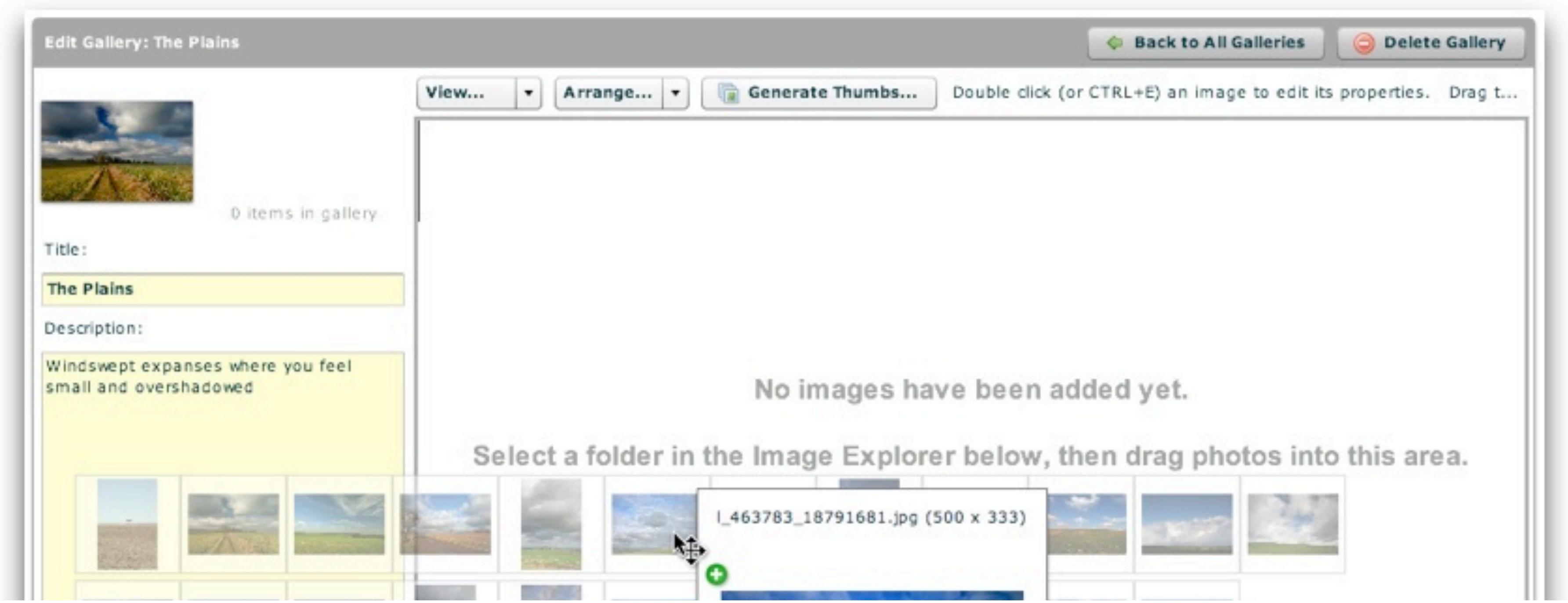
Sort By: (Default)

□ Reverse Order

There are 51 images in the current folder view; use CTRL or SHIFT to select multiple.

Thumb Size:

To add the selected images to the gallery, simply drag and drop them from the Image Explorer into the images grid above:



Edit Gallery: The Plains

0 items in gallery

Title: The Plains

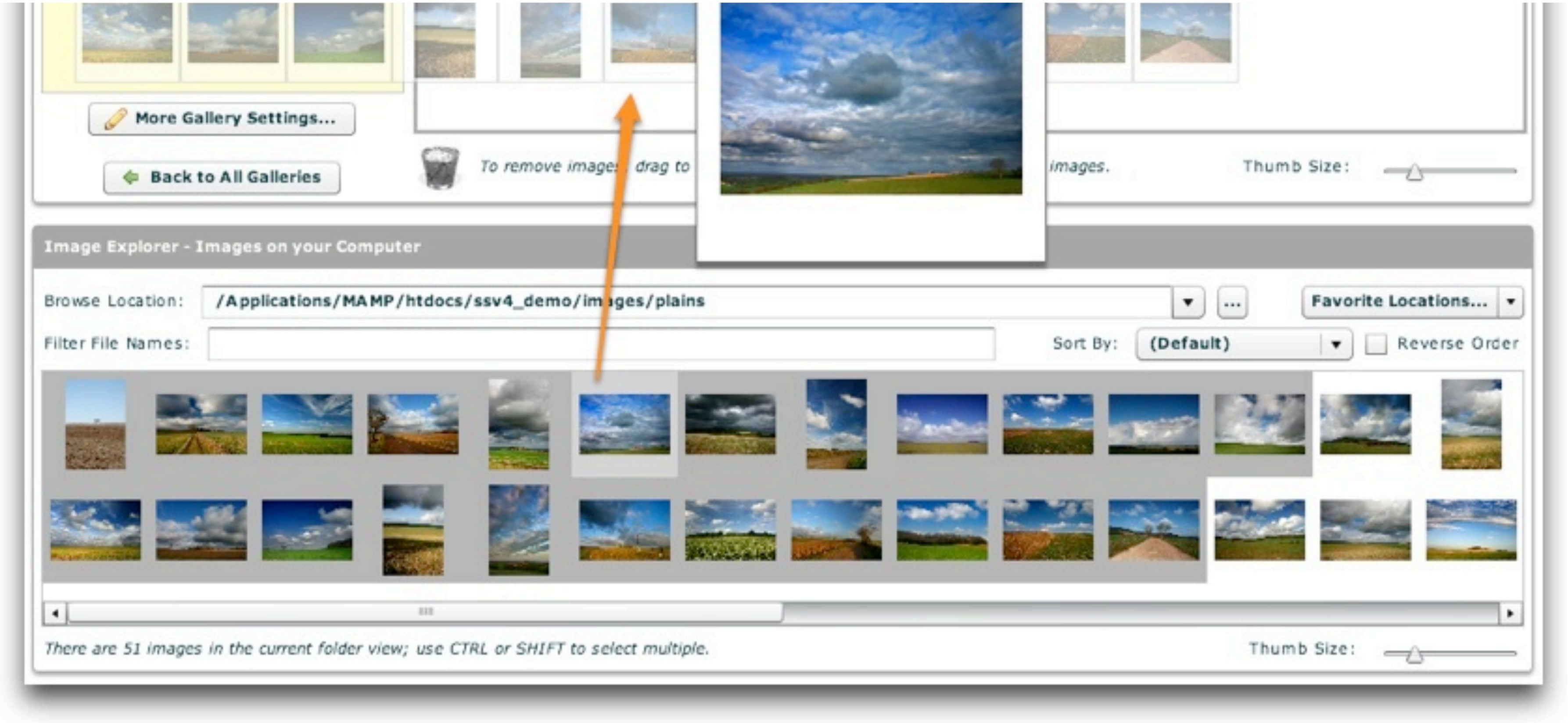
Description: Windswept expanses where you feel small and overshadowed

Double click (or CTRL+E) an image to edit its properties. Drag t...

No images have been added yet.

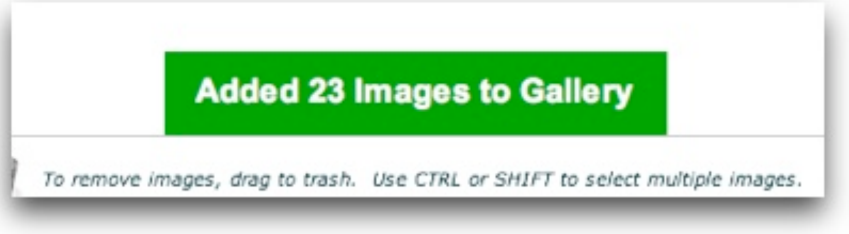
Select a folder in the Image Explorer below, then drag photos into this area.

l_463783_18791681.jpg (500 x 333)



There are 51 images in the current folder view; use CTRL or SHIFT to select multiple.

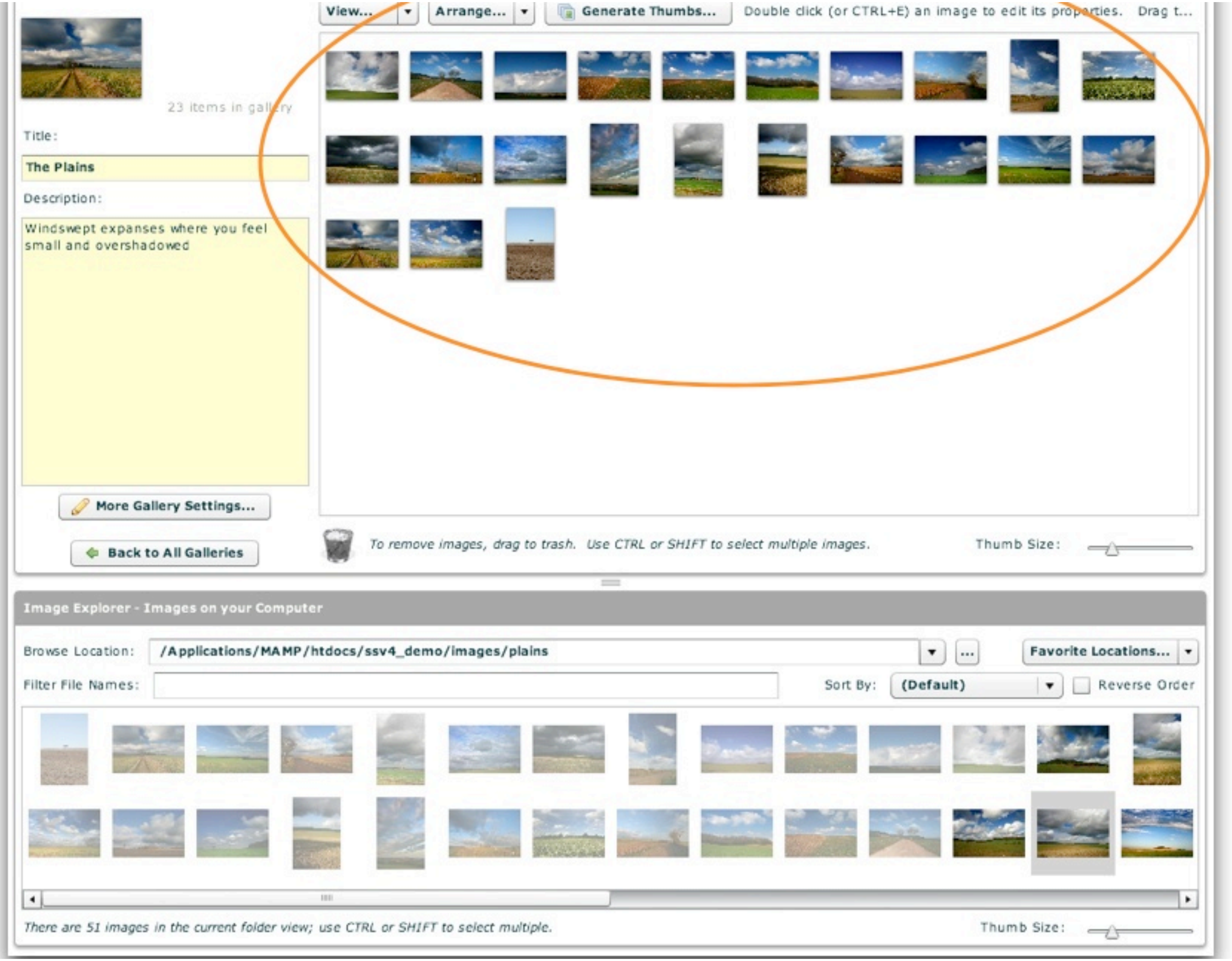
A notification will be displayed, informing you that images were added successfully:



Added 23 Images to Gallery

To remove images, drag to trash. Use CTRL or SHIFT to select multiple images.

And the newly-added images should appear in the images grid:



Double click (or CTRL+E) an image to edit its properties. Drag t...

23 items in gallery

Title: The Plains

Description: Windswept expanses where you feel small and overshadowed

To remove images, drag to trash. Use CTRL or SHIFT to select multiple images.

Thumb Size:

Image Explorer - Images on your Computer

Browse Location: /Applications/MAMP/htdocs/ssv4_demo/images/plains

Filter File Names:

Sort By: (Default)

□ Reverse Order

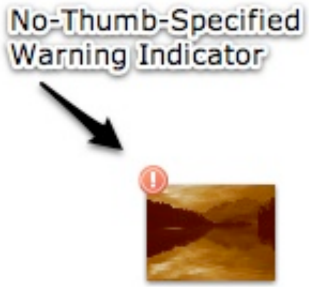
There are 51 images in the current folder view; use CTRL or SHIFT to select multiple.

Thumb Size:

The images which were just added will be grayed-out in the Image Explorer pane, to indicate that they already exist in the gallery.

Important note about images: While you can use images of any size, it is highly recommended that before adding the images to your slideshow you resize them using a graphics program such as Fireworks or Photoshop to be close to the size at which you plan for them to be displayed. If you don't do this, the images will (unnecessarily) take longer to load, reducing slideshow performance and the quality of user experience.

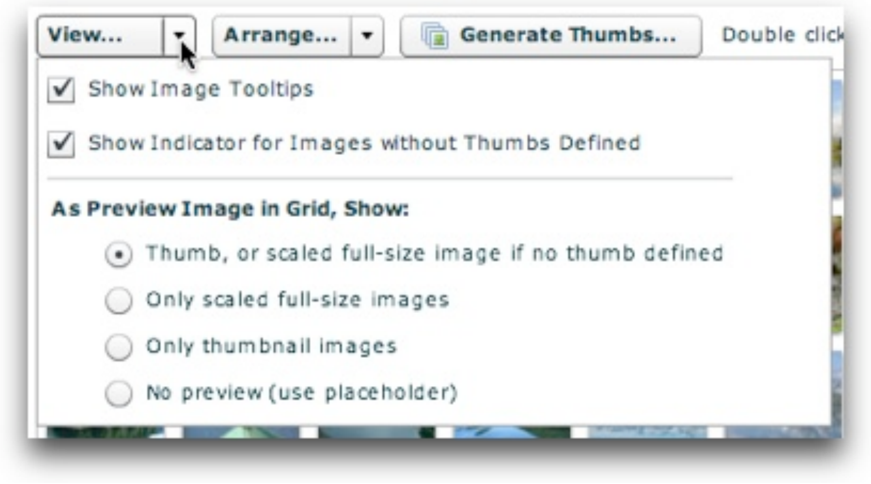
When an image is added and no thumbnail has yet been created for it, a warning indicator will be displayed on the image:



This notifies you that a thumbnail still needs to be created for this image. To learn more about generating thumbnails, see the 'Generating Thumbnail Images' section below. To disable the indicator, see the immediately following 'Customing the Images Grid View' section.

Customizing the Images Grid View (including Tooltips and Thumbnail Indicator)

By default, the Images Grid shows a tile-based grid of all of the images in the gallery. Additionally, when no thumbnail has been specified for an image, a warning indicator is shown in the upper-left corner of that image; and whenever you hover over an image in the Images Grid, a tooltip with image information and a larger preview is displayed. These options can be customized through the 'View...' menu above the images grid:



☑ Show Image Tooltips

☑ Show Indicator for Images without Thumbs Defined

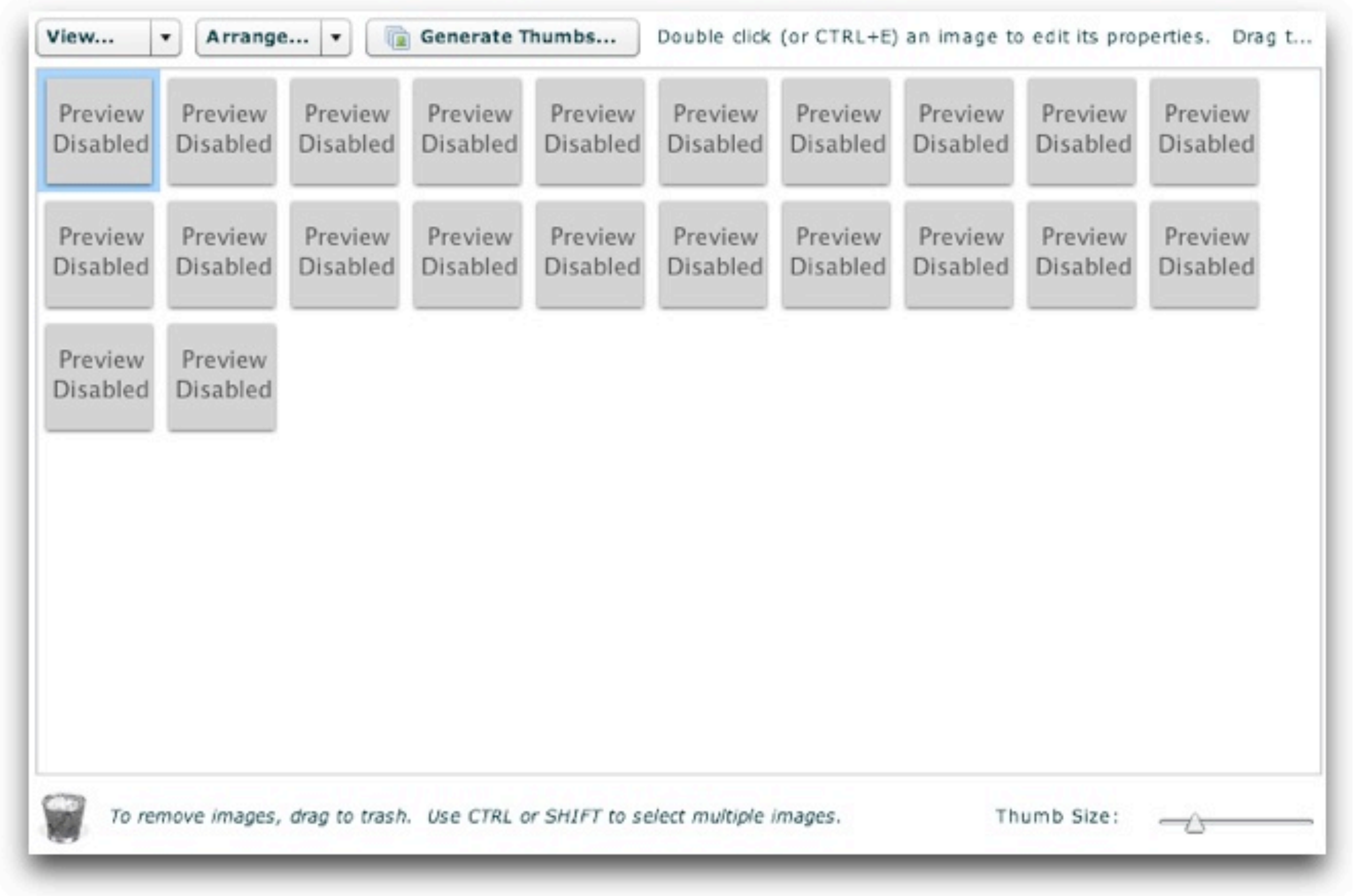
As Preview Image in Grid, Show:

- ● Thumb, or scaled full-size image if no thumb defined
- ○ Only scaled full-size images
- ○ Only thumbnail images
- ○ No preview (use placeholder)

The 'Show Image Tooltips' checkbox enables and disables the image information tooltips.

The 'Show Indicator for Images without Thumbs Defined' checkbox enables and disables the no-thumb-specified warning indicator icon in the upper-left corner of the images in the grid.

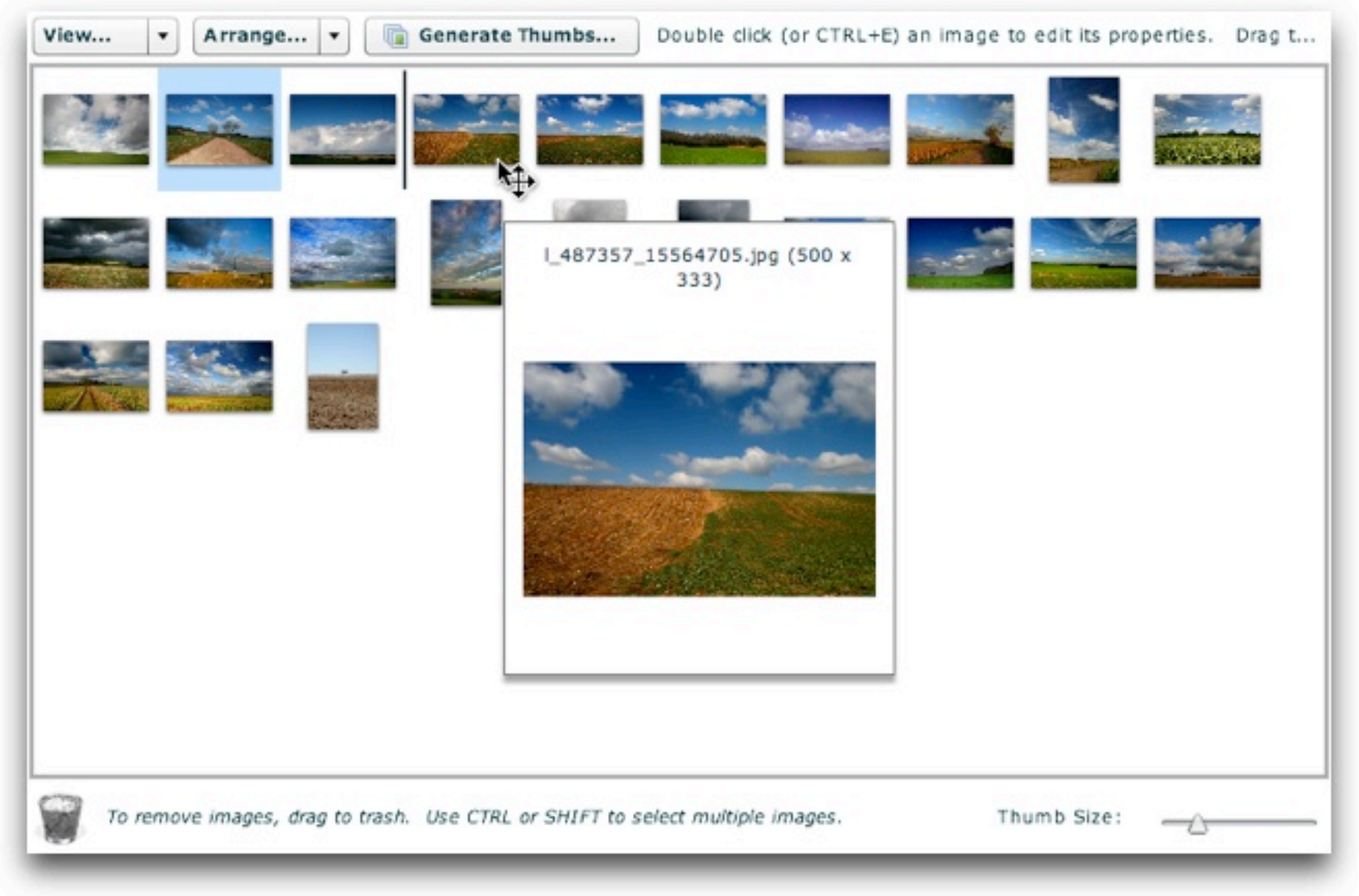
The 'Default Preview Image in Grid' option specifies the default image representation which will be shown in the grid. The default option, 'Thumb, or scaled full-size image if no thumb defined,' will display the image's thumbnail if one is available, or the full-size image if no thumb is available. If 'Only Scaled Full-Size Images' is selected, the full-size images will be loaded and displayed. This option is generally *not* recommended because it could cause sluggish wizard performance and won't provide the best rendering. Using 'Only Thumbnail Images' will display the image's thumbnail if one is available, or a 'No Thumb Specified' placeholder if no thumb is available. This option can be helpful when determining which images still need to have thumbnails created for them. If you wish to entirely disable the preview images and use a placeholder instead, select 'No Preview (Use Placeholder)'. The placeholder option will give the following rendering:


View...
Arrange...
Generate Thumbs...
Double click (or CTRL+E) an image to edit its properties. Drag t...
Preview Disabled
To remove images, drag to trash. Use CTRL or SHIFT to select multiple images.
Thumb Size:

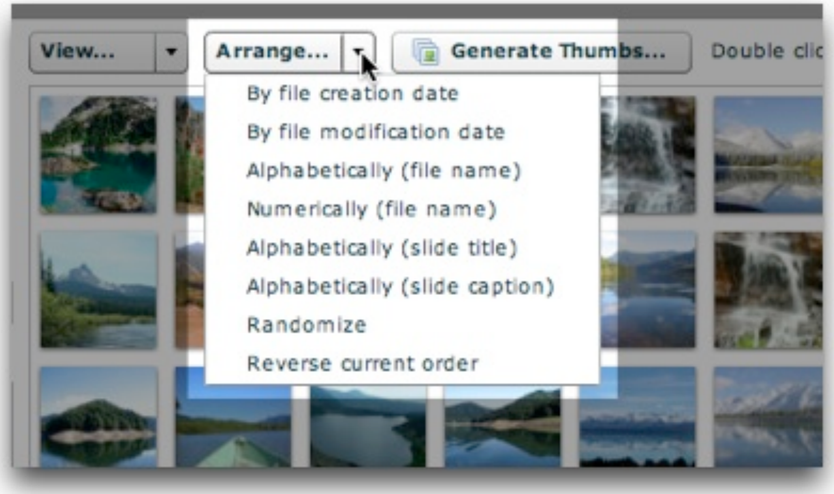
Note: Keep in mind that all of these settings only apply to the wizard; they in no way affect the appearance of your actual slideshow.

Image Management: Reordering, Duplicating and Removing Images

To re-order the images, simply drag the images as needed until the desired order is achieved:


View...
Arrange...
Generate Thumbs...
Double click (or CTRL+E) an image to edit its properties. Drag t...
l_487357_15564705.jpg (500 x 333)
To remove images, drag to trash. Use CTRL or SHIFT to select multiple images.
Thumb Size:

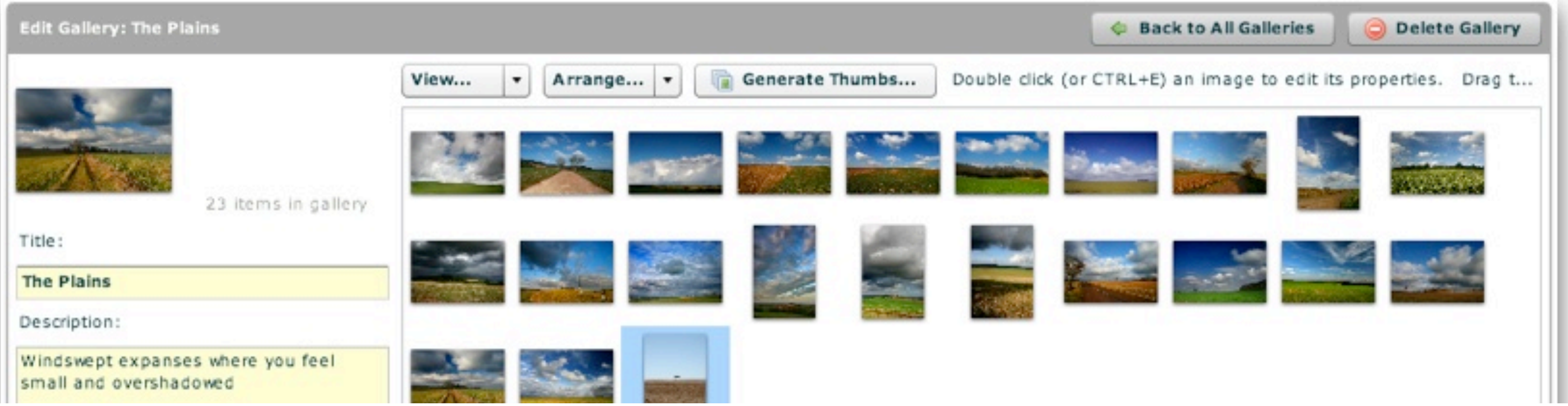
You can also use the 'Arrange...' menu above the grid to perform a quick re-ordering:



Note that all of the options in this menu -- including Randomize -- only randomize the order of the images in the grid. In other words, this is only a design-time randomization. For true randomization at runtime, you must enable randomization in the main Global Configuration / Playback Settings tab. The 'Alphabetically (slide title)' and 'Alphabetically (slide caption)' options can sort the images based on the image caption and title values you have specified for each image in the Image Properties dialog. The 'Numerically (file name)' option sorts based on the first group of numbers found in the file name, similar to the default sort on many operating systems.

To duplicate an image, hold down the CTRL (CMD on Mac) key while dragging the image. When it is dropped, it will be duplicated instead of moved.

To remove an image, drag it to the trash can just below the images grid:

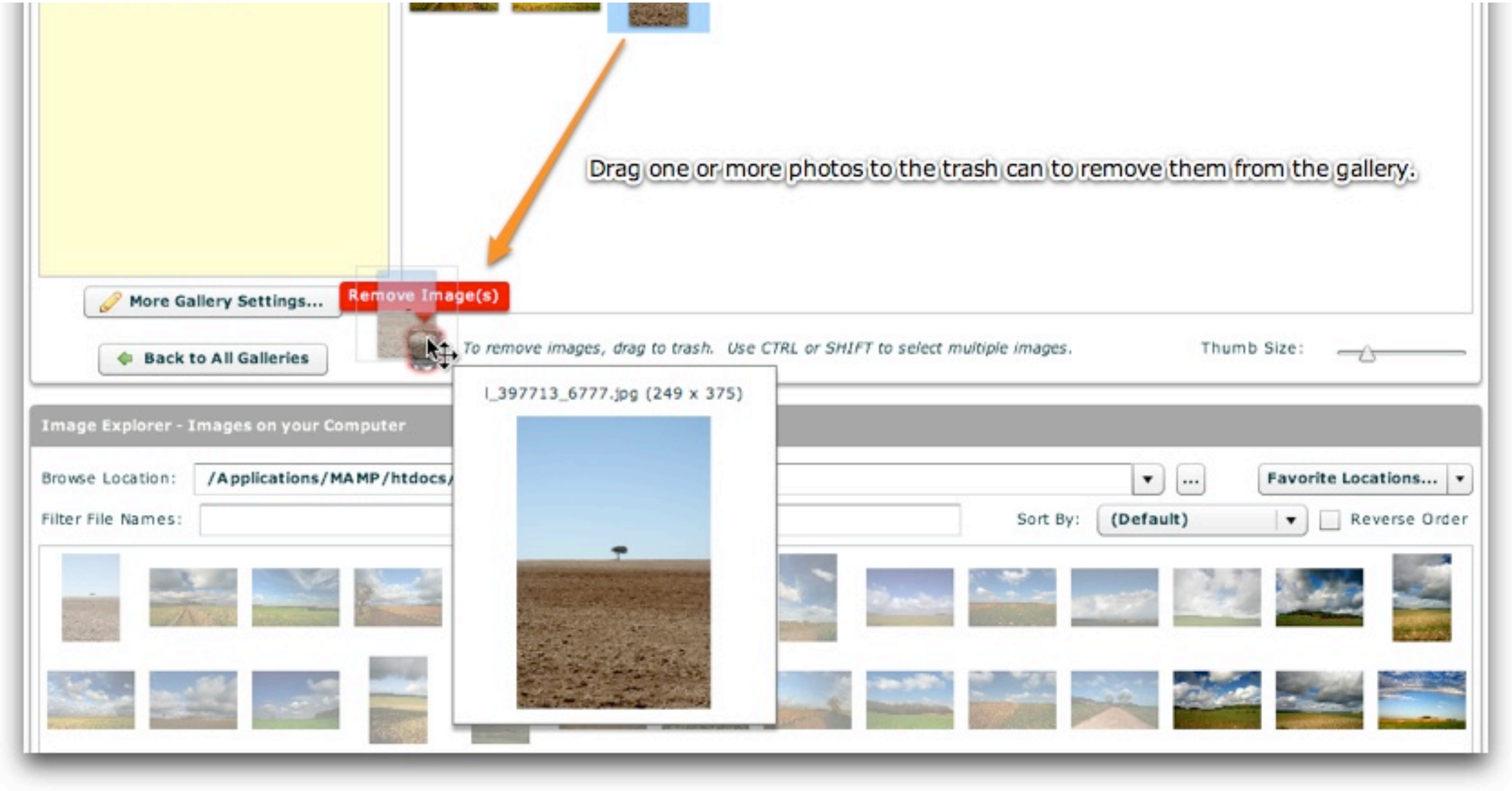


Double click (or CTRL+E) an image to edit its properties. Drag t...

23 items in gallery

Title: The Plains

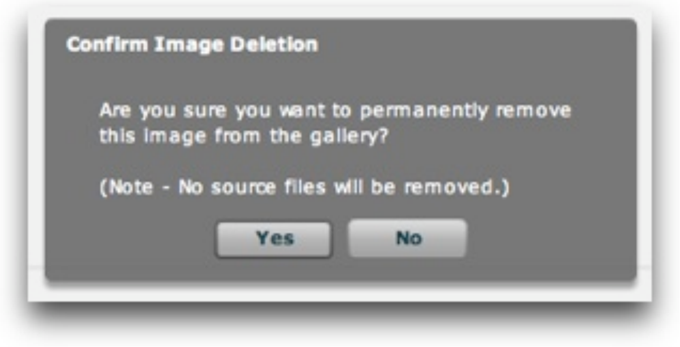
Description: Windswept expanses where you feel small and overshadowed



Drag one or more photos to the trash can to remove them from the gallery.

More Gallery Settings...
Remove Image(s)
Back to All Galleries
To remove images, drag to trash. Use CTRL or SHIFT to select multiple images.
Thumb Size:
l_397713_6777.jpg (249 x 375)
Image Explorer - Images on your Computer
Browse Location:
/Applications/MAMP/htdocs/
...
Favorite Locations...
Filter File Names:
Sort By:
(Default)
Reverse Order

You will be asked to confirm the deletion:

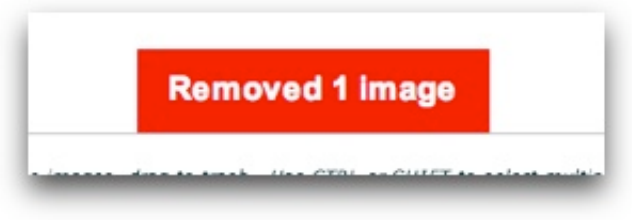


Confirm Image Deletion

Are you sure you want to permanently remove this image from the gallery?

(Note - No source files will be removed.)

As noted in the message, removing an image from the gallery cannot be instantaneously 'undone'.   Any image-specific properties which were set will be lost. Also, keep in mind that the slideshow's configuration is the only thing modified when images are removed; no source files are modified. Thus, you can re-add the images at a later time if desired.   If you allow the image removal, a notification will be displayed confirming the removal:



Removed 1 image

To remove or re-order **more than one image**, you must select multiple images before dragging dropping them.   Just as in the Image Explorer pane, you will use the CTRL (CMD on Macintosh) or SHIFT keys to select multiple images. The CTRL key allows you to select multiple images one-by-one, whereas the SHIFT key allows you to select all images between the current selection and the place where you click with the SHIFT key down.   You can then use the CTRL key to modify the selection if you want to exclude some images within the selected block.   Once you have made a selection, the selection can be dragged in the same manner as which a single image can be dragged.

Generating Thumbnail Images for the Gallery

Now that you have added images to the gallery, you'll want to generate thumbnail images using the Thumbnail Generator. This process is detailed in the Using the Thumbnail Generator subtopic.

Editing the Individual Images' Properties

Once you've added images to the gallery, you'll likely want to set some image-specific properties for each image. These changes are made within the Image Properties dialog, which is accessed by either double-clicking on an image in the images grid, or by selecting it and hitting CTRL+E (CMD+E on Mac).   You can learn more about using the Image Properties dialog in the Editing Image Properties subtopic.

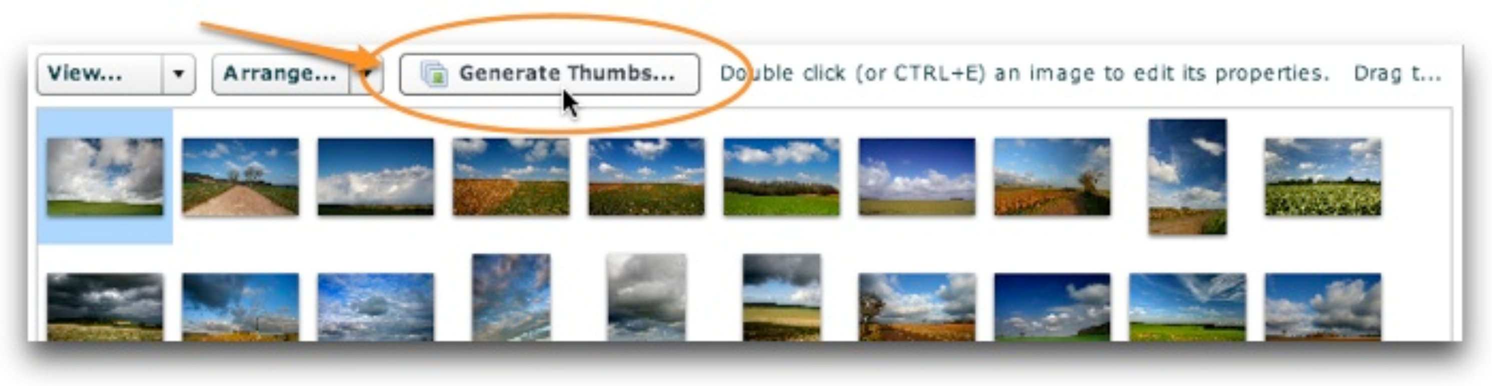
Using the External Wizard to Create and Edit Slideshows > Section-by-Section Usage Guide > Using the 'Images / Galleries' Section > The Gallery Management Pane > Adding Images to a Gallery / Image Management > **Using the Thumbnail Generator**

Using the Thumbnail Generator

One of the most helpful new features in version 4 is the powerful built-in thumbnail generator. This feature allows you to quickly generate thumbnail images for a gallery from within the wizard.

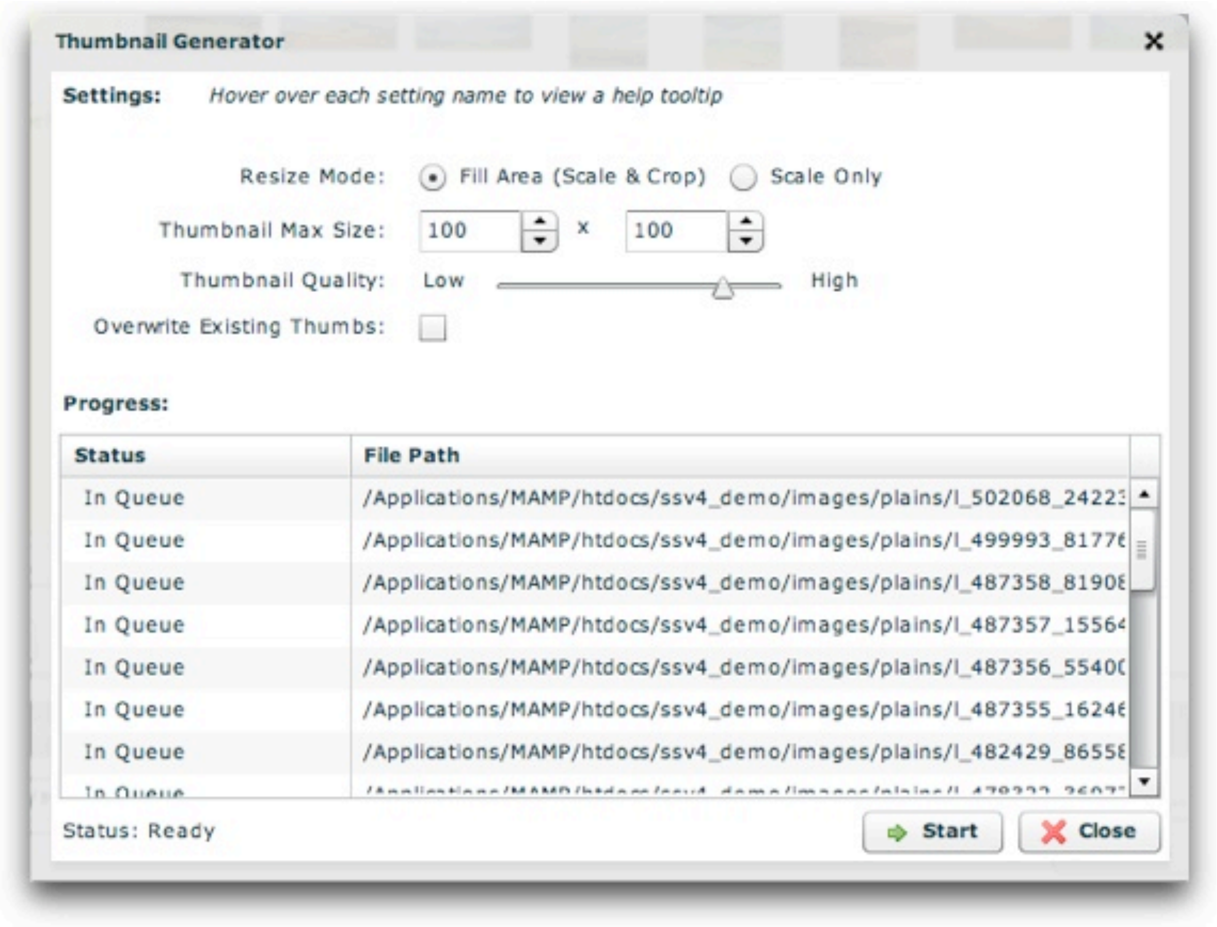
Important note about thumbnails: If you don't specify separate thumbnail images, the full-size images will be loaded and scaled to be used as thumbnails. However, this is **not** an optimal situation, because all images will have to be loaded for a preview to be shown (resulting in lengthy loading times), and the amount of scaling which would occur (> 50%) would result in a pixelated appearance. Thus, to provide a good user experience, it is critical to generate separate thumbnail images and use them with the slideshow.

To access the Thumbnail Generator dialog, you will click the 'Generate Thumbnails...' button just above the images grid in single-gallery-edit view:



Double click (or CTRL+E) an image to edit its properties. Drag t...

The Thumbnail Generator dialog appears as follows:



Thumbnail Generator

Settings: *Hover over each setting name to view a help tooltip*

Resize Mode: ● Fill Area (Scale & Crop) ○ Scale Only

Thumbnail Max Size: 100 x 100

Thumbnail Quality: Low — High

Overwrite Existing Thumbs: □

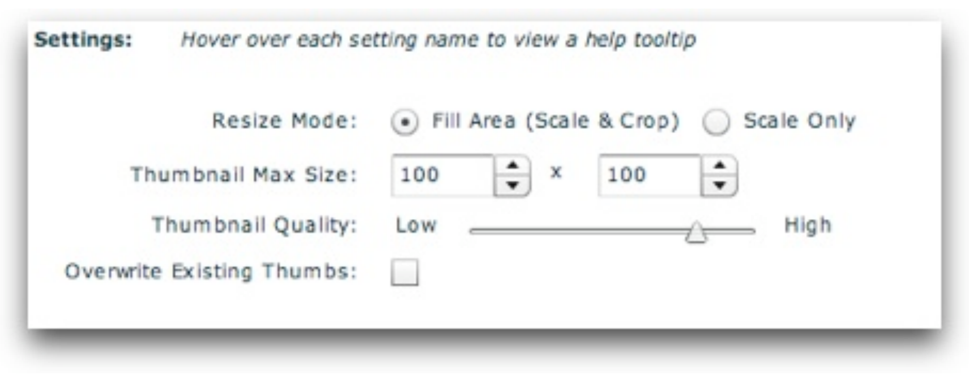
Progress:

Status	File Path
In Queue	/Applications/MAMP/htdocs/ssv4_demo/images/plains/l_502068_24223
In Queue	/Applications/MAMP/htdocs/ssv4_demo/images/plains/l_499993_81776
In Queue	/Applications/MAMP/htdocs/ssv4_demo/images/plains/l_487358_81908
In Queue	/Applications/MAMP/htdocs/ssv4_demo/images/plains/l_487357_15564
In Queue	/Applications/MAMP/htdocs/ssv4_demo/images/plains/l_487356_55400
In Queue	/Applications/MAMP/htdocs/ssv4_demo/images/plains/l_487355_16246
In Queue	/Applications/MAMP/htdocs/ssv4_demo/images/plains/l_482429_86558
In Queue	/Applications/MAMP/htdocs/ssv4_demo/images/plains/l_478333_3607

Status: Ready

All of the images in the gallery which are accessible to the wizard (i.e. they are stored locally on your hard drive, not on the internet) will be automatically added to the progress grid at the bottom.

At the top of the dialog, you will need to review the generator configuration settings:



Settings: *Hover over each setting name to view a help tooltip*

Resize Mode: ● Fill Area (Scale & Crop) ○ Scale Only

Thumbnail Max Size: 100 x 100

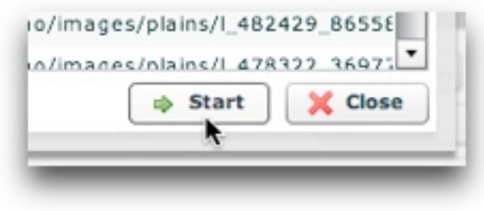
Thumbnail Quality: Low —— High

Overwrite Existing Thumbs: □

Each of the options controls the following (you can view these descriptions directly in the wizard by hovering over the option names):

Property	Description
Resize Mode	Specifies how the images are resized. If you choose 'Fill Area,' the images will be scaled down and then cropped so that they fully fill the specified maximum dimensions. If you choose 'Scale Only,' the images will be scaled down so that they fit entirely within the specified maximum dimensions.
Thumbnail Max Size	Specifies the maximum dimensions, in pixels, for each of the thumbnails. The boxes represent width (the first box) and height (the second box). Keep in mind that the larger the thumbnail size, the larger the file size and the longer the loading times. Thus, you should keep the thumbs relatively small.
Thumbnail Quality	Specifies the quality of the generated thumbnail JPEG files. Usually the default setting (80% quality) is a good setting. Higher quality will make the thumbnails look better, but will result in greater file size and longer loading times. Lower quality sacrifices appearance to achieve smaller file size and faster loading. The quality gains from 80% to 100% are only slightly greater than 80% to 90%, but the file size increase is proportionally much higher. Therefore, if you wish to increase quality beyond 80% it is best to jump to 90% instead of 100%.
Overwrite Existing Thumbs	Specifies if any previously-generated thumbnails should be re-created. If you have **changed any settings**, you should **enable this option**. Generally, you should always enable this option unless the gallery contains hundreds of images and you only want to generate thumbs for a few newly-added images.

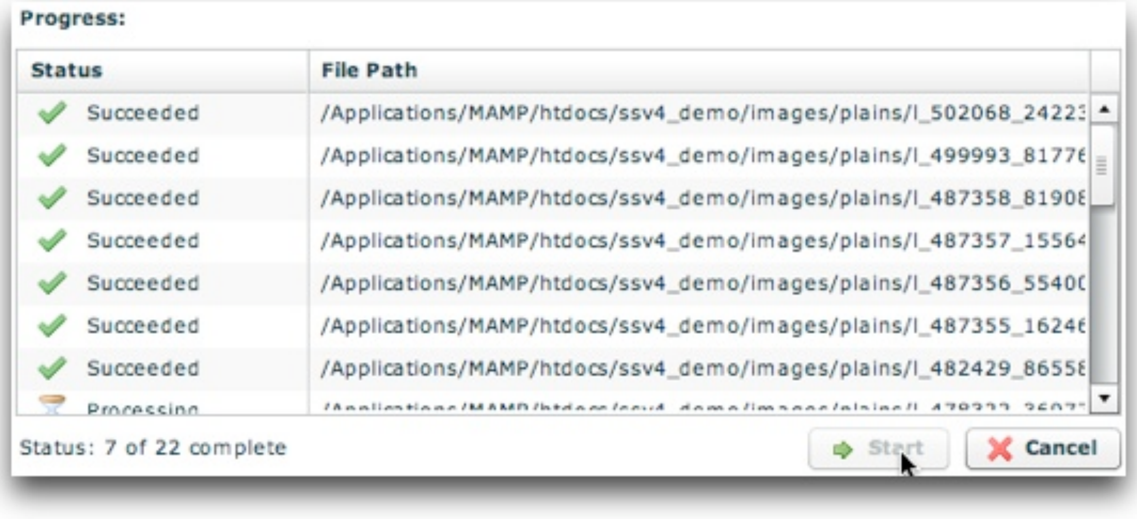
Once you've configured the settings, click the 'Start' button to begin processing:



o/images/plains/l_482429_8655ε

o/images/plains/l_478322_3697

The processing will take place very quickly, usually completing within a few seconds. You will be able to track the progress in the grid:

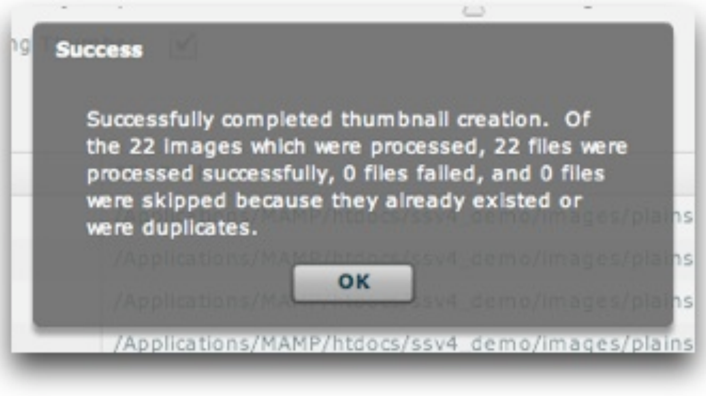


Progress:

Status	File Path
✓ Succeeded	/Applications/MAMP/htdocs/ssv4_demo/images/plains/l_502068_2422
✓ Succeeded	/Applications/MAMP/htdocs/ssv4_demo/images/plains/l_499993_8177
✓ Succeeded	/Applications/MAMP/htdocs/ssv4_demo/images/plains/l_487358_8190
✓ Succeeded	/Applications/MAMP/htdocs/ssv4_demo/images/plains/l_487357_1556
✓ Succeeded	/Applications/MAMP/htdocs/ssv4_demo/images/plains/l_487356_5540
✓ Succeeded	/Applications/MAMP/htdocs/ssv4_demo/images/plains/l_487355_1624
✓ Succeeded	/Applications/MAMP/htdocs/ssv4_demo/images/plains/l_482429_8655
Processing	/Applications/MAMP/htdocs/ssv4_demo/images/plains/l_478322_3697

Status: 7 of 22 complete

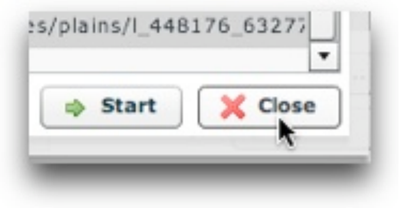
If you need to stop the processing for some reason, you can click the Cancel button. When processing completes, you will be notified of the results:



Success

Successfully completed thumbnail creation. Of the 22 images which were processed, 22 files were processed successfully, 0 files failed, and 0 files were skipped because they already existed or were duplicates.

Now that the thumbnails have been generated, you can click the Close button to return to the main wizard:



Note: When thumbnails are generated, each image is automatically updated to use the new thumbnail; you never have to worry about entering thumbnail URL's. Each thumbnail is automatically placed in an `xfs_thumbs/` subfolder within the image's folder, using the same name as the original image.

Editing Image Properties

Double-clicking on an individual image brings you to the Image Properties dialog. This dialog is where you will configure all of the image-specific properties such as the title, caption, and image link. The dialog appears as follows:

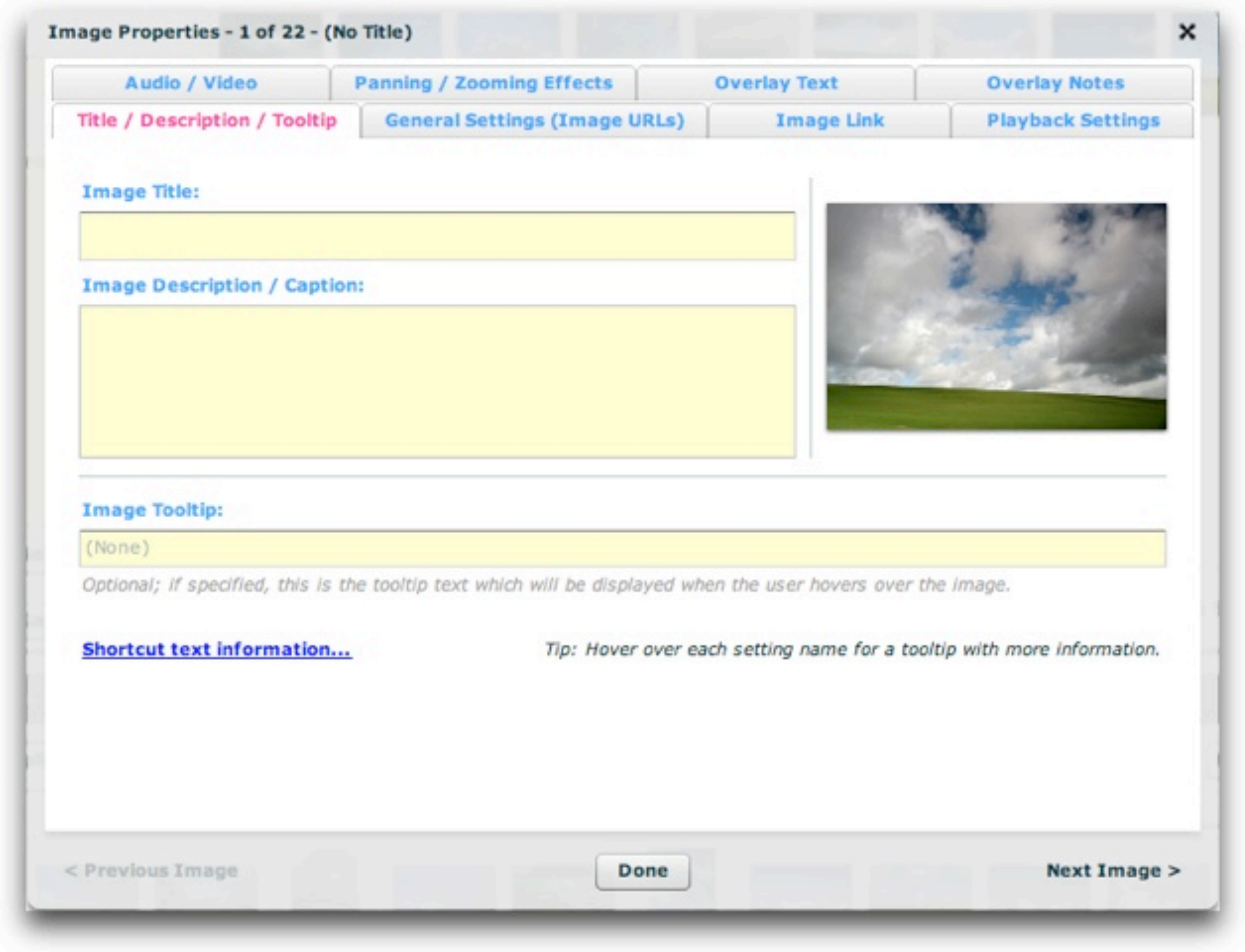


Image Properties - 1 of 22 - (No Title)

Audio / Video
Panning / Zooming Effects
Overlay Text
Overlay Notes
Title / Description / Tooltip
General Settings (Image URLs)
Image Link
Playback Settings

Image Title:

Image Description / Caption:

Image Tooltip:

(None)

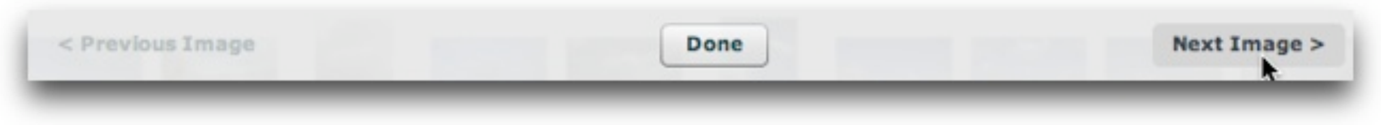
Optional; if specified, this is the tooltip text which will be displayed when the user hovers over the image.

Shortcut text information...

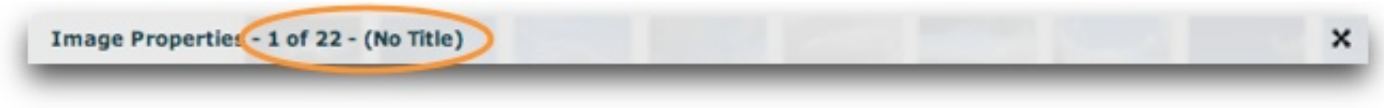
Tip: Hover over each setting name for a tooltip with more information.

To navigate the dialog, click on the tabs at the top of the window. Each of the tabs are individually covered in this topic's subtopics. See these subtopics for detailed information.

When you're done making changes, click the **Done** button at the bottom to close the Image Properties dialog. If you want to edit the properties of multiple images more easily, you can click the '< Previous Image' or 'Next Image >' buttons at the bottom of the window; this will make the dialog switch to the previous or next image for editing. Pressing **CTRL + >** or **CTRL + <** (use CMD on Mac) has the same effect as well.

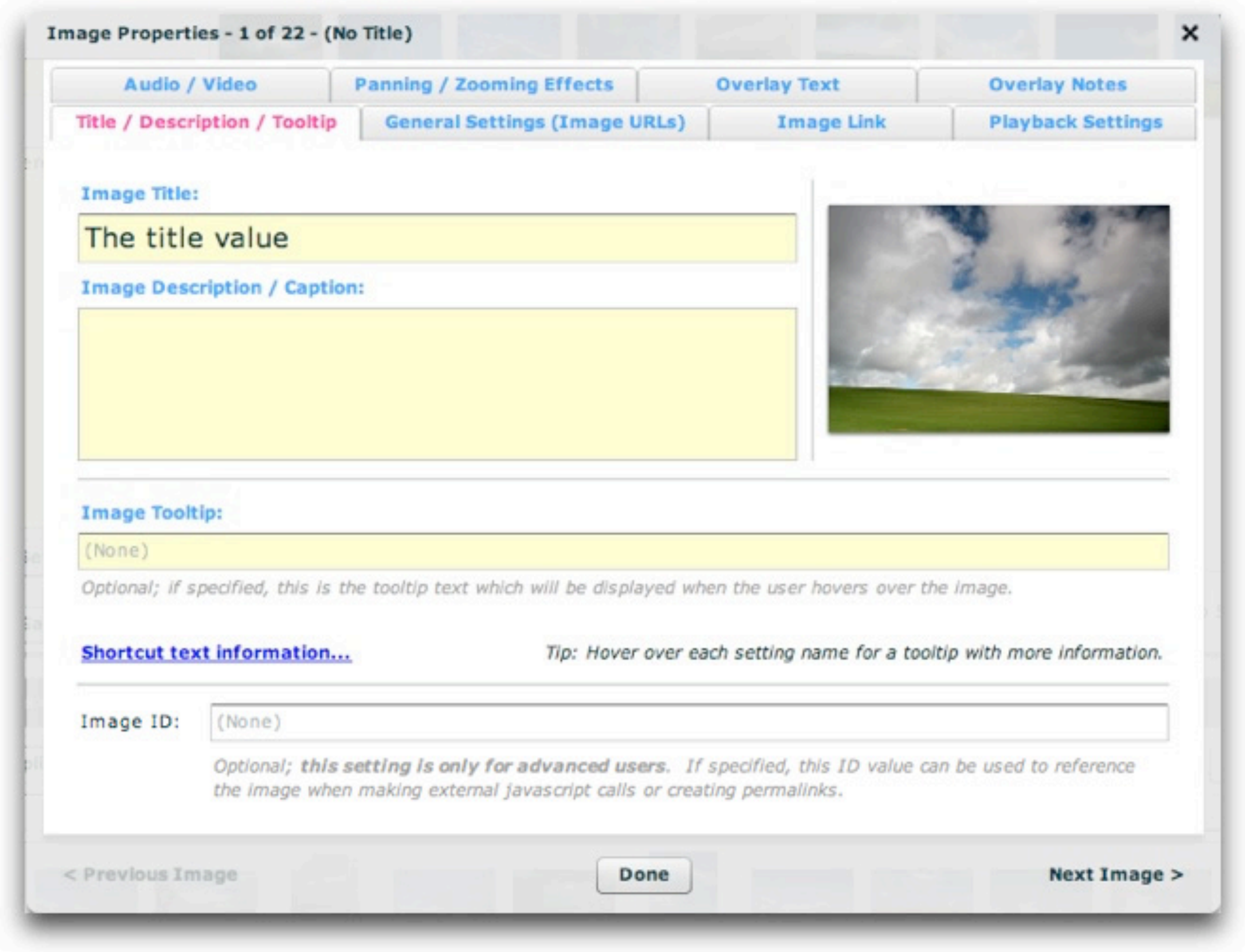


The index and title of the image are displayed in the dialog title:



Title / Description / Tooltip

The Title / Description / Tooltip tab is the most commonly used tab. This is where you specify the title and description/caption for the image. The image tooltip and reference ID can also be specified in this section. The tab appears as follows:



×
Audio / Video
Panning / Zooming Effects
Overlay Text
Overlay Notes
Title / Description / Tooltip
General Settings (Image URLs)
Image Link
Playback Settings
Shortcut text information...
< Previous Image
Done
Next Image >

Image Properties - 1 of 22 - (No Title)

Image Title: The title value

Image Description / Caption:

Image Tooltip: (None)

Optional; if specified, this is the tooltip text which will be displayed when the user hovers over the image.

Tip: Hover over each setting name for a tooltip with more information.

Image ID: (None)

Optional; ***this setting is only for advanced users****. If specified, this ID value can be used to reference the image when making external javascript calls or creating permalinks.*

The most important values are the title and caption/description, which are described in more detail below. To the right of these settings, you'll see the image preview, which offers a quick preview of the image in question:

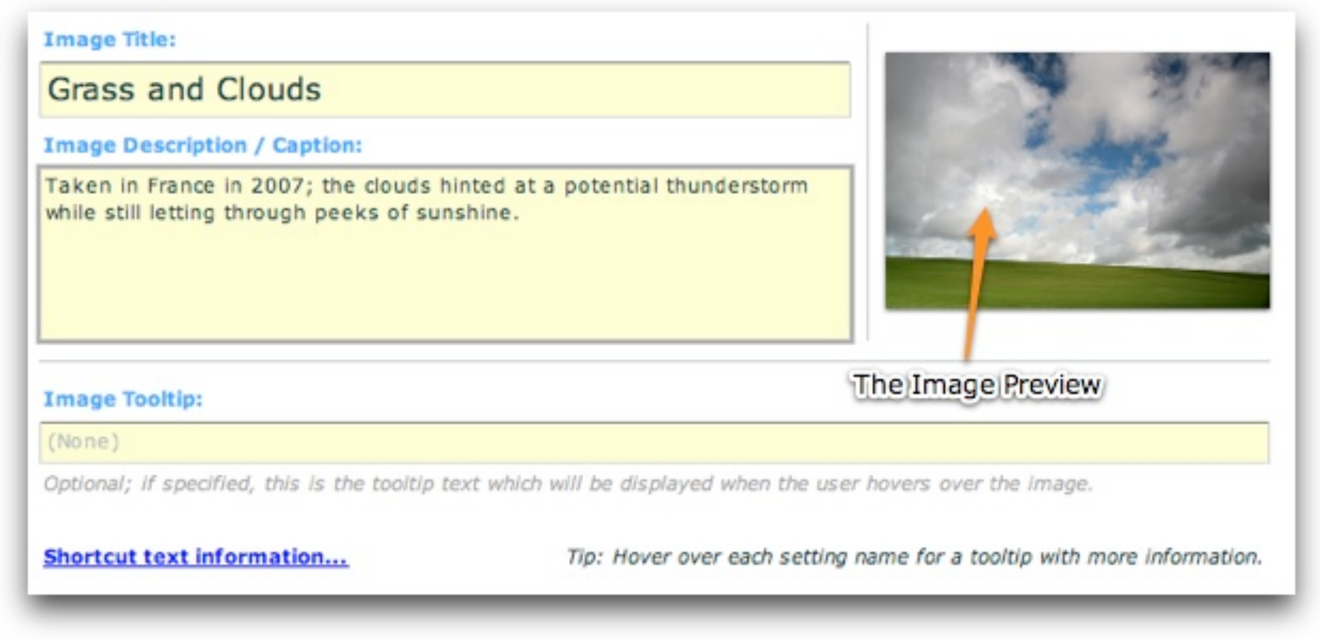


Image Title:
Grass and Clouds

Image Description / Caption:
Taken in France in 2007; the clouds hinted at a potential thunderstorm while still letting through peeks of sunshine.

Image Tooltip:
(None)

Optional; if specified, this is the tooltip text which will be displayed when the user hovers over the image.

Shortcut text information...

Tip: Hover over each setting name for a tooltip with more information.

About Shortcut Text (AKA Replacement Values)

Before covering all of the options, it is important to note the availability of shortcut text (also known as "replacement values") which can be used in certain fields to reference values in other fields.

In the title, description, link, link target, and tooltip fields, you can add a shortcut to other properties and attribute values on the image.

The following image-related replacement values are available:

- `{image.index}` or `{image.id}` - This string is replaced with the 1-base index of the current image. For example, "This is image #{image.id}" would become "This is image #1" while the first image was open, then subsequently "This is image #2", etc.
- `{image.total}` - This string is replaced with the total number of images in the current gallery. For example, "This is image {image.id}/{image.total}" would become "This is image 1/10" if there were ten images in the gallery and the first image was open.
- `{image.link}` - This string is replaced with the current image's link value (if one has been defined). For example, "Purchase from: {image.link}" would become "Purchase from: http://www.mysite.com/" if the image's link was defined like this: `<img link="http://www.mysite.com/" ... />`.
- `{image.title}` - This string is replaced with the current image's title value (if one has been defined). For example, "Current image: {image.title}"

would become "Current image: Daffodil" if the image's title was defined like this: `<img ... title="Daffodil" ... />`.

- `{image.caption}` - This string is replaced with the current image's caption value (if one has been defined). For example, "About: {image.caption}" would become "About: Taken two weeks ago on holiday in South Africa" if the image's caption was defined like this: `<img ... caption="Taken two weeks ago on holiday in South Africa" ... />`.
- `{image.`*`someOtherAtt`*`}` - This string is replaced with the corresponding XML attribute of the current image's XML node. For example, "Current image: {image.title}" would become "Current image: Blue Orchid" if the image's title was defined like this: <img title="Blue Orchid" ... >. You can use the standard attributes (such as title and caption), or any custom attribute which has been defined. In other words, {image.someValue} would be replaced with `<img ... someValue="Special attribute value!" ... />`
- **[Metadata support]** `{metadata.`*`property`*`}` - This string is replaced with various available metadata parameters that are read from the loaded images. Click here to learn more about metadata integration. The available *property* values include `{metadata.creator}` (corresponds to the Creator metadata property), `{metadata.title}` (corresponds to the Title metadata property), and `{metadata.description}` (corresponds to the Description metadata property). If a requested metadata property doesn't exist on an image, or if the metadata isn't available because the image hasn't fully loaded yet, a default value can be specified using the following format: `{metadata.property`**`|Default Value Here`**`}` . After the property value, a pipe character denotes the start of the default string. For example, the following value would display *"Loading..."* until the metadata title became available: `{metadata.title|Loading...}` .

Other less-common image-related replacement values are available too:

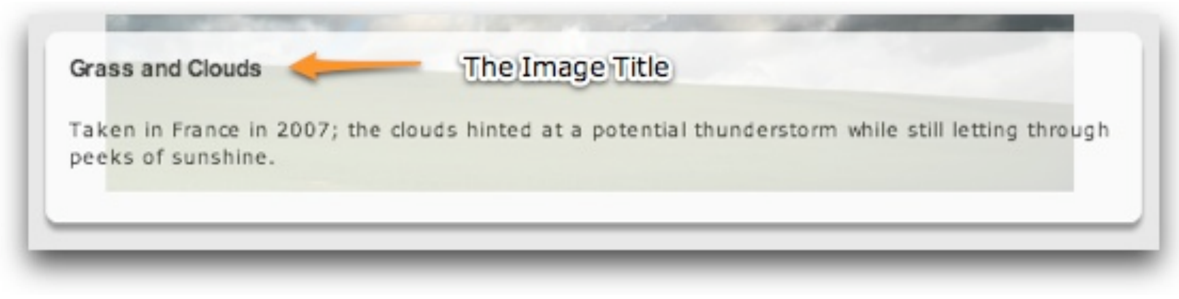
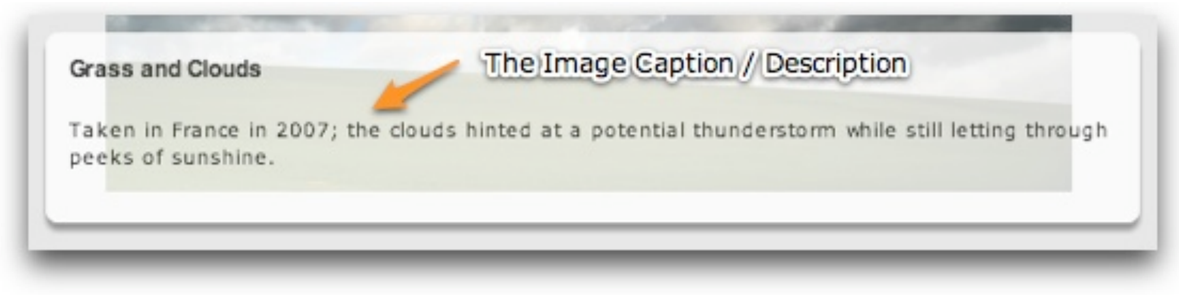
- `{image.index0}` or `{image.id0}` - This string is replaced with the **0**-base index of the current image. This is generally a property used only by advanced users; for most purposes, the 1-base `{image.index}` value is appropriate.
- `{image.uniqID}` - This string is replaced with the uniqID index of the current image. This can be used for permalinking, and will not be affected by randomization. This is generally a property only used by advanced users.
- `{image.rawID}` - This string is replaced with the id attribute defined on the actual image node; this is the value set in the wizard in the Image Linkage ID field. This is generally a property only used by advanced users.
- `{image.src}` - This string is replaced with the current image's src value. This usually represents just the file name of the image; if you want to get the entire image path, use the {image.srcFull} replacement value.
- `{image.srcFull}` - This string is replaced with the current image's full image path (use {image.src} for just the shorter path / file name).
- `{image.thumbFull}` - This string is replaced with the current image's full thumb path (use {image.thumb} for just the shorter path / file name)

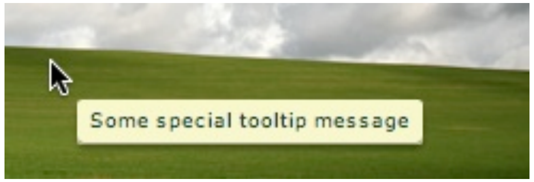
For all replacement values, you can use the following format to specify a default value: `{image.property`**`|Default Value Here`**`}` . Follow the property name with a pipe character (`|`) and the default value before the closing curly brace.

Note that you can use multiple replacement values in the same string.

Option Overview

This table covers all of the options in this tab:

Property	Description
Image Title	This is the title associated with the image. Along with the description/caption, these are generally the two most commonly used (and important) properties you'll set for each image. This is the title value which will be displayed in the slideshow's Caption element, or in any dynamic text element for `{image.title}` .  Grass and Clouds Taken in France in 2007; the clouds hinted at a potential thunderstorm while still letting through peeks of sunshine.
Image Description / Caption	This is the description (or caption) associated with the images. Along with the title, these are generally the two most commonly used (and important) properties you'll set for each image. This is the description/caption value which will be displayed in the slideshow's Caption element, or in any dynamic text element for `{image.caption}` .  Grass and Clouds Taken in France in 2007; the clouds hinted at a potential thunderstorm while still letting through peeks of sunshine.
Image Tooltip	This is the tooltip text which will be displayed when a user hovers over the image. For example:

	 Some special tooltip message Leaving the default blank value disables the tooltip. **Note for advanced users:** To customize the appearance of the tooltip, use the `HTMLToolTip` selector within the styles section of your XML configuration file. See the com.dwuser.ss4.ui.extras.HTMLToolTip class for more information about available properties. Note for advanced users: To references this value in other locations using shortcut text (replacement values), use `{image.toolTip}` .
Image ID	This **optional** setting is only for advanced users. It specifies a unique ID by which can be used to reference the image when creating permalinks or making javascript interface calls. As described in permalinks topic, specifying an ID makes it simple to link directly to a specific image in a slideshow. Note: Only letters, numbers and underscores may be used in this value.

XML Equivalents (for Advanced Users and Developers)

The settings above map to the following XML attributes on each `<img ... />` node:

Property	Equivalent <img ... /> Attribute
Image Title	`@title`
Image Description / Caption	`@caption`
Image Tooltip	`@toolTip`
Image ID	`@id`

Note that the '@' is part of customary syntactical representation, and is not part of each attribute name.

For more detailed XML information, see the Developer Reference.

General Settings (Image URL's)

The 'General Settings (Image URLs)' tab is where you can manually specify or modify the image paths. The tab appears as follows:

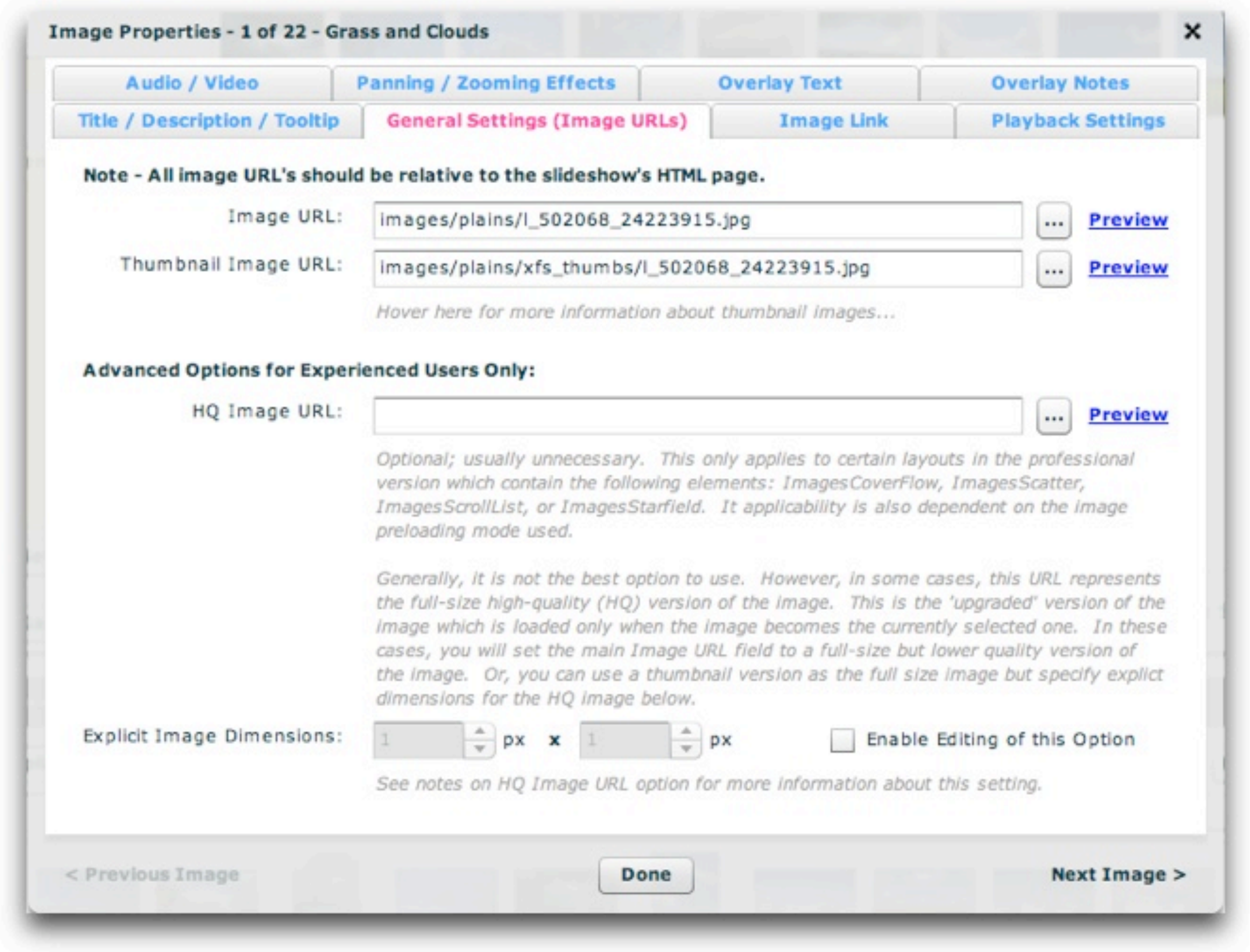


Image Properties - 1 of 22 - Grass and Clouds

×
Audio / Video
Panning / Zooming Effects
Overlay Text
Overlay Notes
Title / Description / Tooltip
General Settings (Image URLs)
Image Link
Playback Settings

Note - All image URL's should be relative to the slideshow's HTML page.

Image URL: images/plains/l_502068_24223915.jpg ... Preview

Thumbnail Image URL: images/plains/xfs_thumbs/l_502068_24223915.jpg ... Preview

Hover here for more information about thumbnail images...

Advanced Options for Experienced Users Only:

HQ Image URL: ... Preview

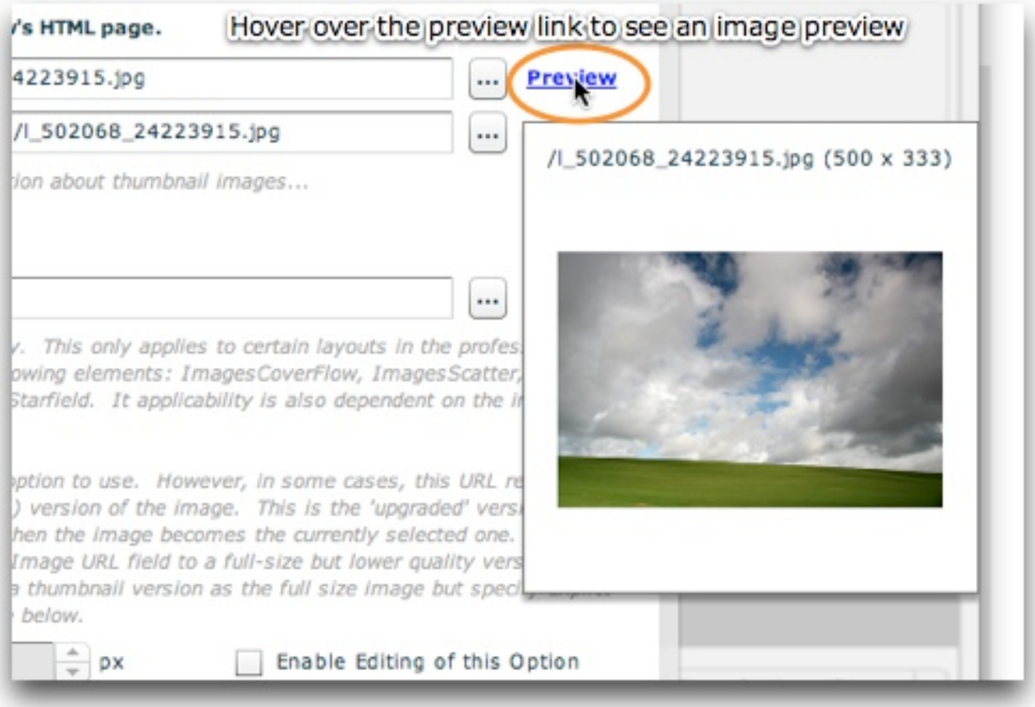
Optional; usually unnecessary. This only applies to certain layouts in the professional version which contain the following elements: ImagesCoverFlow, ImagesScatter, ImagesScrollList, or ImagesStarfield. It applicability is also dependent on the image preloading mode used.

Generally, it is not the best option to use. However, in some cases, this URL represents the full-size high-quality (HQ) version of the image. This is the 'upgraded' version of the image which is loaded only when the image becomes the currently selected one. In these cases, you will set the main Image URL field to a full-size but lower quality version of the image. Or, you can use a thumbnail version as the full size image but specify explict dimensions for the HQ image below.

Explicit Image Dimensions: 1 px x 1 px □ Enable Editing of this Option

See notes on HQ Image URL option for more information about this setting.

You can preview any of the images by hovering over the 'Preview' link to the right of the URL box. A popup tooltip with a larger preview, path, and image dimensions will be displayed:


Hover over the preview link to see an image preview

's HTML page.

4223915.jpg

/l_502068_24223915.jpg

ion about thumbnail images...

y. This only applies to certain layouts in the profes
owing elements: ImagesCoverFlow, ImagesScatter,
Starfield. It applicability is also dependent on the i

ption to use. However, in some cases, this URL re
) version of the image. This is the 'upgraded' vers
hen the image becomes the currently selected one.
Image URL field to a full-size but lower quality vers
a thumbnail version as the full size image but speci
below.

/l_502068_24223915.jpg (500 x 333)

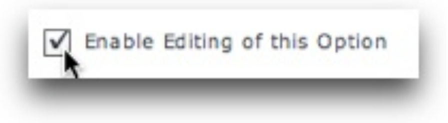
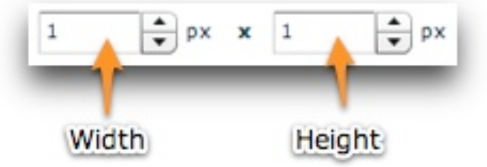
px ☐ Enable Editing of this Option

Option Overview

First, it is important to note that all of the URLs in this section **must be relative to the HTML page holding the slideshow**. Otherwise, the paths will fail. You can ensure that the paths are correct by letting the wizard generate all paths by using the browse ('...') buttons.

Additionally, it is important to note that all images must be within your Dreamweaver site, or they will not be accessible. While not required, best practices dictate storing them all in a common `images/` folder.

Property	Description
Image URL	(Required) This is the URL of the slide's image. This can be any valid image URL, pointing to a JPEG, GIF, PNG, or even an SWF animation. Keep in mind that whenever you load from a remote site you may run into Flash Player Cross-domain Loading Restrictions.
Thumbnail Image URL	(Optional, but recommended) This can be any valid image URL, pointing to a JPEG, GIF, PNG, or even an SWF animation. Keep in mind that whenever you load from a remote site you may run into Flash Player Cross-domain Loading Restrictions. If you omit this parameter, the full-size image will be loaded, scaled down, and used as the thumbnail. However, it is **highly**

	recommended that you create separate thumbnail images because using the full-size images often results in lengthly loading times, delayed previews, and pixelation due to the high level of scaling. Using separate thumbnail images ensures an all-around better user experience. You can generate thumbnails automatically for the entire gallery by using the Thumbnail Generator (covered previously in detail in the linked topic).
HQ (High Quality) Image URL	*(Advanced parameter for experienced users only)* (Optional; usually unnecessary) This only applies to certain layouts in the professional version which contain one of the following image display elements: ImagesCoverFlow, ImagesScatter, ImagesScrollList, or ImagesStarfield. Its applicability is also dependent on the image preloading mode used (especially the 3D preloading mode). Generally, it is not the best option to use. However, in some cases, this URL represents the full-size high-quality (HQ) version of the image. This is the 'upgraded' version of the image which is loaded only when the image becomes the currently selected one. In these cases, you will set the main Image URL field to a full-size but lower quality (hence smaller file size and quicker loading) version of the image. Or, you can use a thumbnail version as the full size image but specify explicit dimensions for the HQ image below. These dimensions will be used to scale the thumb up as if it were the full-size image, displaying it until the HQ image loads.
Explicit Image Dimensions	*(Advanced parameter for experienced users only)* If using the HQ Image URL as described above, in some cases you must specify the explicit dimensions of the full-size image so the slideshow knows before the image actually loads (see HQ Image description for more info). To prevent inexperienced users from straying into this area unnecessarily, you must select the 'Enable Editing of this Option' checkbox to make edits:  ☑ Enable Editing of this Option Then, you can enter explicit dimensions. The two boxes are the width and height respectively. Both are denominated in pixels:  1 px x 1 px

Advanced Techniques

At some point, you might want to directly add a remote image to your gallery (an image which is not on your local hard drive). To do this, you must first add a temporary placeholder image from your local drive. Then, you can open the Image Properties dialog, switch to this tab, and manually enter the remote URL. Keep in mind that whenever you load from a remote site you may run into Flash Player Cross-domain Loading Restrictions.

XML Equivalents (for Advanced Users and Developers)

The settings above map to the following XML attributes on each `<img ... />` node:

Property	Equivalent <img ... /> Attribute
Image URL	`@src`
Thumbnail Image URL	`@thumb`
HQ (High Quality) Image URL	`@hq`
Explicit Image Width	`@w`
Explicit Image Height	`@h`

Note that the '@' is part of customary syntactical representation, and is not part of each attribute name.

For more detailed XML information, see the Developer Reference.

Image Link

The Image Link tab of the Image Properties dialog is where you will specify the link which is attached to the image. This is the page which is jumped to when the image is clicked. The tab appears as follows:

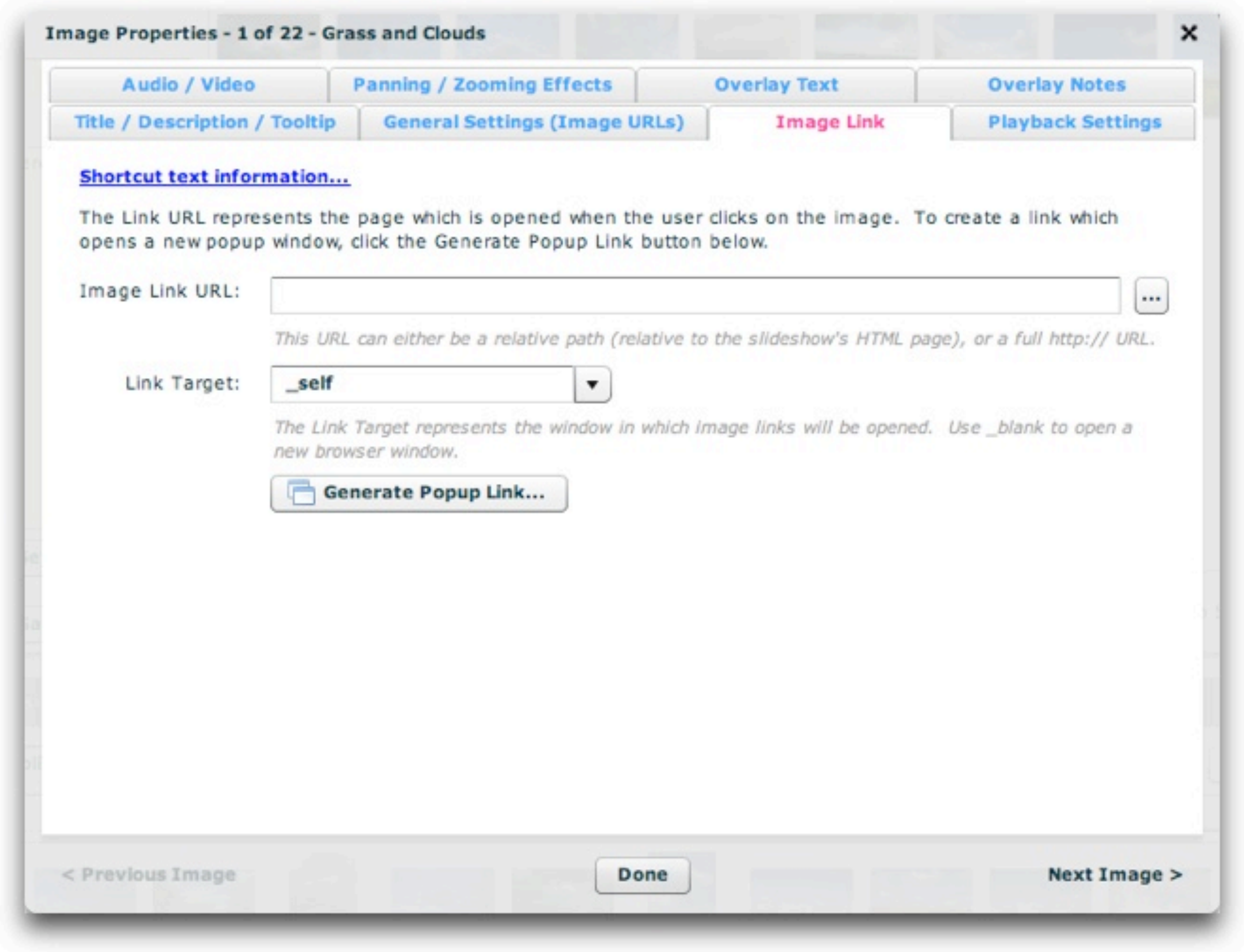


Image Properties - 1 of 22 - Grass and Clouds
Audio / Video
Panning / Zooming Effects
Overlay Text
Overlay Notes
Title / Description / Tooltip
General Settings (Image URLs)
Image Link
Playback Settings

Shortcut text information...

The Link URL represents the page which is opened when the user clicks on the image. To create a link which opens a new popup window, click the Generate Popup Link button below.

Image Link URL:

This URL can either be a relative path (relative to the slideshow's HTML page), or a full http:// URL.

Link Target: _self

The Link Target represents the window in which image links will be opened. Use _blank to open a new browser window.

About Shortcut Text (AKA Replacement Values)

Before covering all of the options, it is important to note the availability of shortcut text (also known as "replacement values") which can be used in certain fields to reference values in other fields.

In the title, description, link, link target, and tooltip fields, you can add a shortcut to other properties and attribute values on the image.

The following image-related replacement values are available:

- `{image.index}` or `{image.id}` - This string is replaced with the 1-base index of the current image. For example, "This is image #{image.id}" would become "This is image #1" while the first image was open, then subsequently "This is image #2", etc.
- `{image.total}` - This string is replaced with the total number of images in the current gallery. For example, "This is image {image.id}/{image.total}" would become "This is image 1/10" if there were ten images in the gallery and the first image was open.
- `{image.link}` - This string is replaced with the current image's link value (if one has been defined). For example, "Purchase from: {image.link}" would become "Purchase from: http://www.mysite.com/" if the image's link was defined like this: `<img link="http://www.mysite.com/" ... />`.
- `{image.title}` - This string is replaced with the current image's title value (if one has been defined). For example, "Current image: {image.title}" would become "Current image: Daffodil" if the image's title was defined like this: `<img ... title="Daffodil" ... />`.
- `{image.caption}` - This string is replaced with the current image's caption value (if one has been defined). For example, "About: {image.caption}" would become "About: Taken two weeks ago on holiday in South Africa" if the image's caption was defined like this: `<img ... caption="Taken two weeks ago on holiday in South Africa" ... />`.
- `{image.`*`someOtherAtt`*`}` - This string is replaced with the corresponding XML attribute of the current image's XML node. For example, "Current image: {image.title}" would become "Current image: Blue Orchid" if the image's title was defined like this: <img title="Blue Orchid" ... >. You can use the standard attributes (such as title and caption), or any custom attribute which has been defined. In other words, {image.someValue} would be replaced with `<img ... someValue="Special attribute value!" ... />`
- **[Metadata support]** `{metadata.`*`property`*`}` - This string is replaced with various available metadata parameters that are read from the loaded images. Click here to learn more about metadata integration. The available *property* values include `{metadata.creator}` (corresponds to the Creator metadata property), `{metadata.title}` (corresponds to the Title metadata property), and `{metadata.description}` (corresponds to the Description metadata property). If a requested metadata property doesn't exist on an image, or if the metadata isn't available because the image hasn't fully loaded yet, a default value can be specified using the following format: `{metadata.property`**`|Default Value Here`**`}` . After the property value, a pipe character denotes the start of the default string. For example, the following value would display *"Loading..."* until the metadata title became available: `{metadata.title|Loading...}` .

Other less-common image-related replacement values are available too:

- `{image.index0}` or `{image.id0}` - This string is replaced with the **0**-base index of the current image. This is generally a property used only by advanced users; for most purposes, the 1-base `{image.index}` value is appropriate.
- `{image.uniqID}` - This string is replaced with the uniqID index of the current image. This can be used for permalinking, and will not be affected by randomization. This is generally a property only used by advanced users.
- `{image.rawID}` - This string is replaced with the id attribute defined on the actual image node; this is the value set in the wizard in the Image Linkage ID field. This is generally a property only used by advanced users.

- {image.src} - This string is replaced with the current image's src value. This usually represents just the file name of the image; if you want to get the entire image path, use the {image.srcFull} replacement value.
- {image.srcFull} - This string is replaced with the current image's full image path (use {image.src} for just the shorter path / file name).
- {image.thumbFull} - This string is replaced with the current image's full thumb path (use {image.thumb} for just the shorter path / file name)

For all replacement values, you can use the following format to specify a default value: {image.property**|Default Value Here**} . Follow the property name with a pipe character (|) and the default value before the closing curly brace.

Note that you can use multiple replacement values in the same string.

One of the most common usage examples is setting the link URL to be {image.src} and the link target to _blank. This would make the full-size image open in a new browser window when it is clicked, allowing the user to view, download or print the original image.

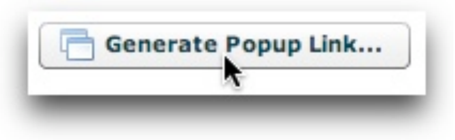
Option Overview

Property	Description
Image Link URL	This is the link which is opened when the user clicks on the image. The URL can be either a relative path (relative to the slideshow's HTML page), or a full http:// URL. You can use the browse button ('...') to the right of the URL field to browse for and select a page; in this case the proper path will be automatically generated. You can also call javascript by using links in the following format: javascript:void(myJavascriptFunction('param1', 'param2')); If you want to have the link open in a true popup window (which gives you control over the size and appearance of the window), you must use a special javascript link. The Popup Link Builder will automatically generate this type of link for you. See the Using the Popup Link Builder section just below for more information. **Important note:** Whenever using a popup link or a javascript link, you should set the Link Target option to _self.
Link Target	The link target represents the window in which the image link will be opened. Specifying a value here will override any gallery-wide defaults established in the Gallery Properties dialog. The default value of _self will open the link in the current window. Using _blank will open the link in a new browser window. Using _parent will open the link in the parent page if using frames. Specifying any other value will control a specific frame in a frameset. For control over the appearance and size of a new window, you must use a true popup window. See the Image Link URL property directly above for more information about this.

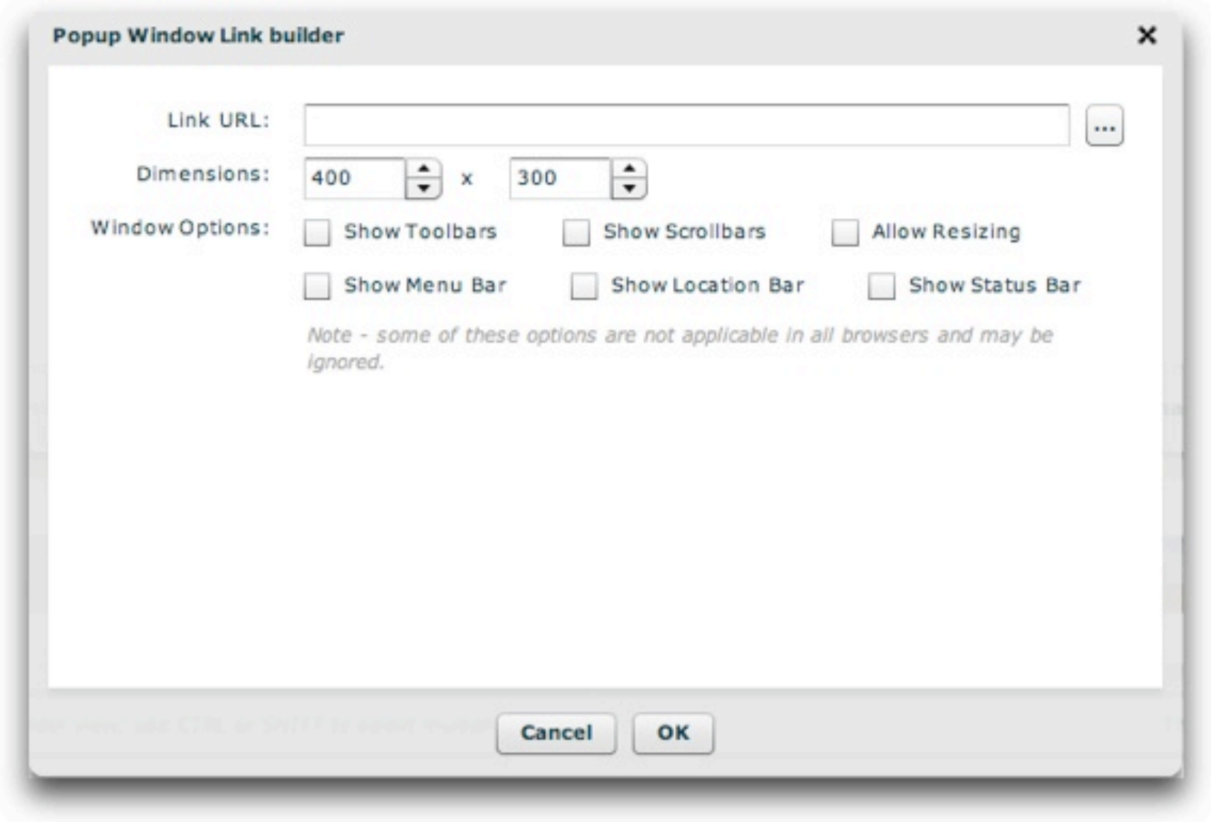
Important note: Whenever using a popup link or a javascript link, you should set the Link Target option to `_self`.

Using the Popup Link Builder

The Popup Link Builder builds the necessary links to open true popup windows when an image is clicked on. This feature is only available in the Pro version of the software - learn more about upgrading here. To get started, click the 'Generate Popup Link...' button:



The Popup Link Builder dialog appears as follows:



Popup Window Link builder

Link URL:

Dimensions: 400 x 300

Window Options: □ Show Toolbars □ Show Scrollbars □ Allow Resizing

□ Show Menu Bar □ Show Location Bar □ Show Status Bar

Note - some of these options are not applicable in all browsers and may be ignored.

Each of the options control the following:

Property	Description
Link URL	This is the URL which you want opened in the popup. The same guidelines apply as those for the Image Link URL propery shown above. You can use the browse button ('...') located to the right of the URL box to browse to and select the page you want to link to.
Dimensions	Specifies the width and height, in pixels, of the popup window which is opened. The first field is the width, the second the height.
Window Options	These settings control the appearance of the popup window. Note that some of the options are ignored by some browsers. By default, the popup window will be almost entirely blank; no toolbars or other chrome will be shown.

When you're done configuring each of the options, click the OK button to generate the link and return to the main window:

Note: Some browsers may require your users to temporarily allow popups to be able to launch popups from the slideshow. You may want to provide a note mentioning this if you rely heavily on popups.

Advanced Techniques

Automatic URL Jumping

You can also configure the slideshow to automatically open the image's link when the image is opened. To do this, you must enable an automatic jump for the Autoplay Delay setting in the Playback Settings tab of the Image Properties dialog. See the topic which covers that tab for more complete information. This feature can be used to create an intro-slideshow which automatically jumps to the main site once it finishes playing (i.e. the last slide is reached).

XML Equivalents (for Advanced Users and Developers)

The settings above map to the following XML attributes on each `<img ... />` node:

Property	Equivalent <img ... /> Attribute
Image Link URL	`@link`
Link Target	`@linkTarget`

Note that the '@' is part of customary syntactical representation, and is not part of each attribute name.

For more detailed XML information, see the Developer Reference.

Playback Settings

The Playback Settings tab is where you will configure playback options specific to this image/slide. All values specified here will override the global defaults set in the main 'Global Configuration and Playback Settings' tab, as well as any gallery-specific defaults established in the Gallery Properties dialog. The tab appears as follows:

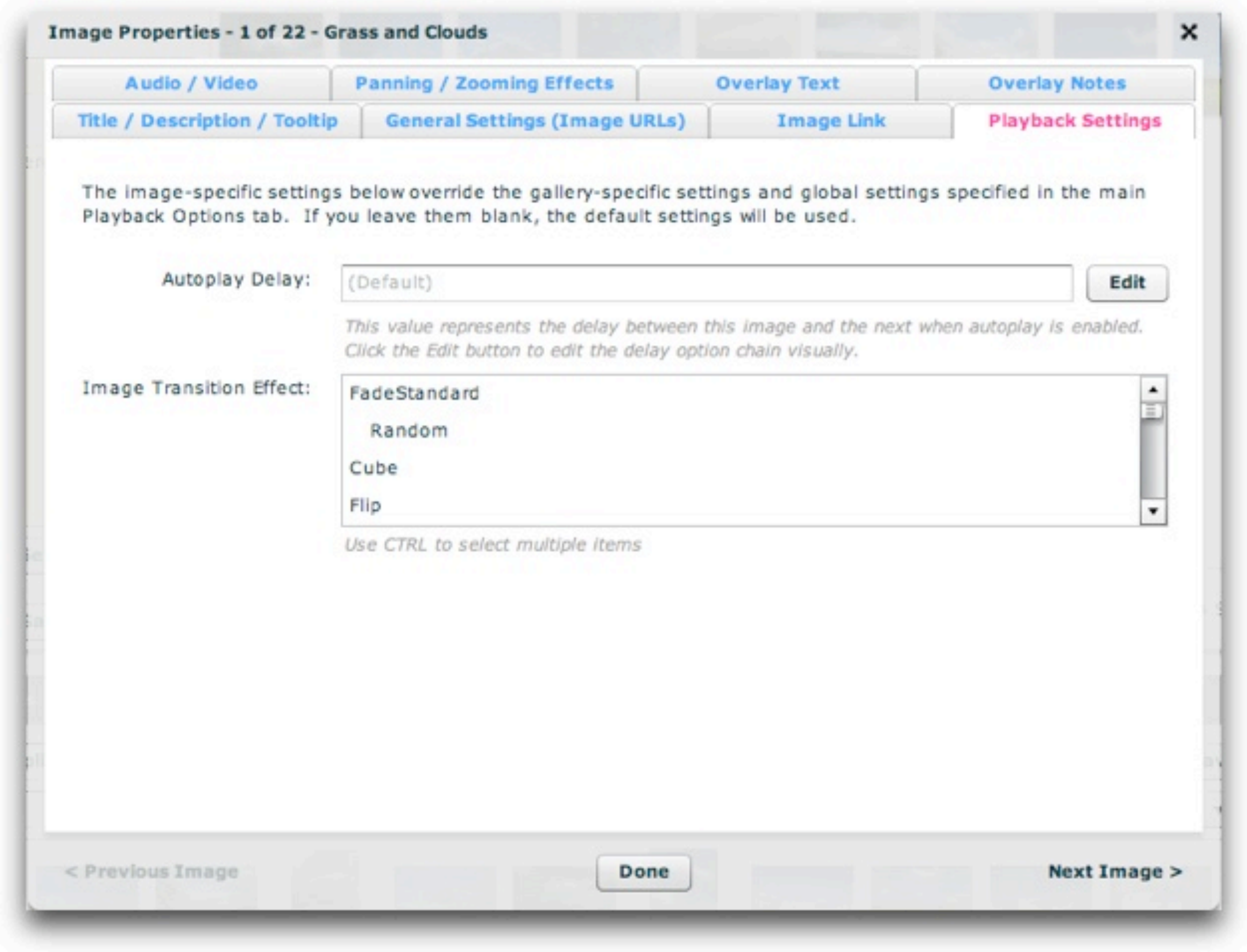


Image Properties - 1 of 22 - Grass and Clouds

The image-specific settings below override the gallery-specific settings and global settings specified in the main Playback Options tab. If you leave them blank, the default settings will be used.

Autoplay Delay: (Default)

This value represents the delay between this image and the next when autoplay is enabled. Click the Edit button to edit the delay option chain visually.

Image Transition Effect:

Use CTRL to select multiple items

The options control the following:

Property	Description
Autoplay Delay	When autoplay is enabled, this value specifies the delay between each image/slide. Because this is a slide-specific value, it configures how long the current image/slide is shown. You can also specify a value which stops autoplay immediately, or automatically jumps to the image link associated with this image. These values are specially constructed strings; clicking the 'Edit' button allows you to visually edit the value. To learn more, see Using the

	Autoplay Delay Editor.
Image Transition Effect	This parameter only applies if you're using the ImagesStandard image display element (most slideshows do). In these cases, you can specify the image transition effect for opening this image, overriding the global setting (specified in the main Global Configuration & Playback Settings tab) as well as any gallery-specific setting (specified in the Gallery Properties dialog). To view a description of each effect, you can hover over the effect name. Or, to view a summary of all effects, see the Setting the Image Transition topic in the *Using the 'Global Configuration and Playback Settings' Section* topic. If you don't wish to have the description tooltips, you can temporarily disable the tooltips by selecting File > Enable/Disable All Tooltips on the program menu. Selecting multiple items (by holding down the CTRL or CMD key) will make the slideshow randomly cycle through those selected transitions. To cycle through *all* of the transitions randomly, select only the **Random** option.

XML Equivalents (for Advanced Users and Developers)

The settings above map to the following XML attributes on each `<img ... />` node:

Property	Equivalent <img ... /> Attribute	Notes
Autoplay Delay	`@autoplayDelay`	To learn more about the `autoplayDelay` format, see the `com.dwuser.ss4.managers.AutoplayManager` class in the API Reference.
Image Transition Effect	`@effectType`	If using more than one effect, each of their names must be separated with a comma (no spaces). For more information about supported transitions, see Setting the Image Transition. For example: `...effectType="FadeStandard,Cube" ...`

Note that the '@' is part of customary syntactical representation, and is not part of each attribute name.

For more detailed XML information, see the Developer Reference.

Using the External Wizard to Create and Edit Slideshows > Section-by-Section Usage Guide > Using the 'Images / Galleries' Section > The Gallery Management Pane > Adding Images to a Gallery / Image Management > Editing Image Properties > **Audio / Video**

Audio / Video

The Audio / Video tab is where you can specify audio or video to be associated with this slide. The tab appears as follows:

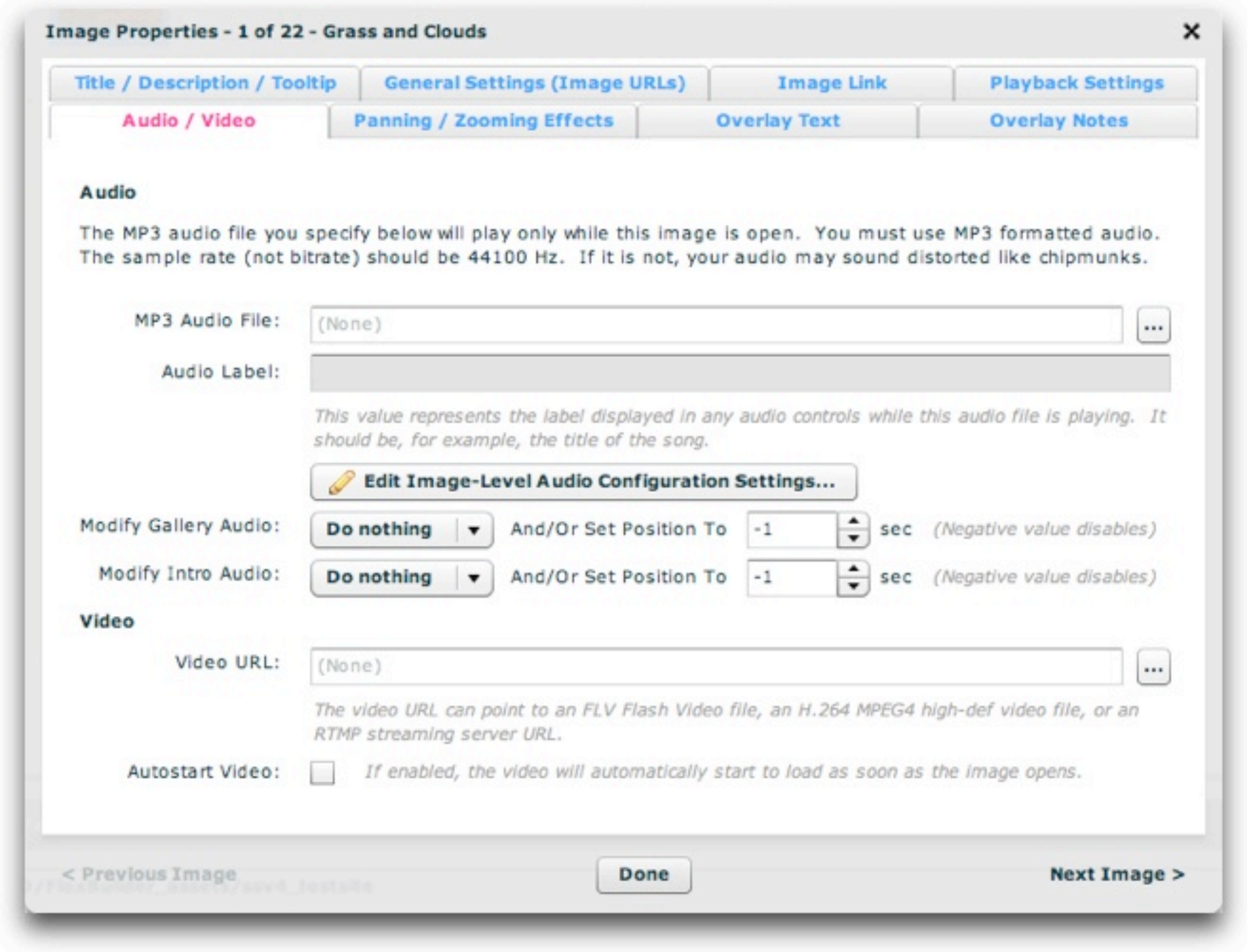


Image Properties - 1 of 22 - Grass and Clouds

Title / Description / Tooltip
General Settings (Image URLs)
Image Link
Playback Settings
Audio / Video
Panning / Zooming Effects
Overlay Text
Overlay Notes

Audio

The MP3 audio file you specify below will play only while this image is open. You must use MP3 formatted audio. The sample rate (not bitrate) should be 44100 Hz. If it is not, your audio may sound distorted like chipmunks.

MP3 Audio File: (None) ...

Audio Label:

This value represents the label displayed in any audio controls while this audio file is playing. It should be, for example, the title of the song.

Modify Gallery Audio: Do nothing ▾ And/Or Set Position To -1 sec *(Negative value disables)*

Modify Intro Audio: Do nothing ▾ And/Or Set Position To -1 sec *(Negative value disables)*

Video

Video URL: (None) ...

The video URL can point to an FLV Flash Video file, an H.264 MPEG4 high-def video file, or an RTMP streaming server URL.

Autostart Video: □ *If enabled, the video will automatically start to load as soon as the image opens.*

Adding Audio

This is where you can specify image-level audio. This is the audio which plays only while this image/slide is open. The audio file must be in MP3 format; for necessary formatting guidelines, see Avoiding "Chipmunks" Audio: MP3 Formatting Guidelines. Enter the URL in the box or use the browse button to the right ('...') to select a file. The URL must be relative to the HTML page which holds the slideshow.

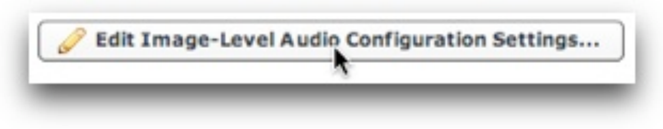
Note: If you are loading audio from a **different domain** than the one on which the slideshow is displayed, you will be affected by Flash Player crossdomain loading security restrictions. To learn more, see this topic.

Note: If you're audio plays back at a **distorted speed**, like "**chipmunks**", you need to review the MP3 formatting guidelines referenced above.

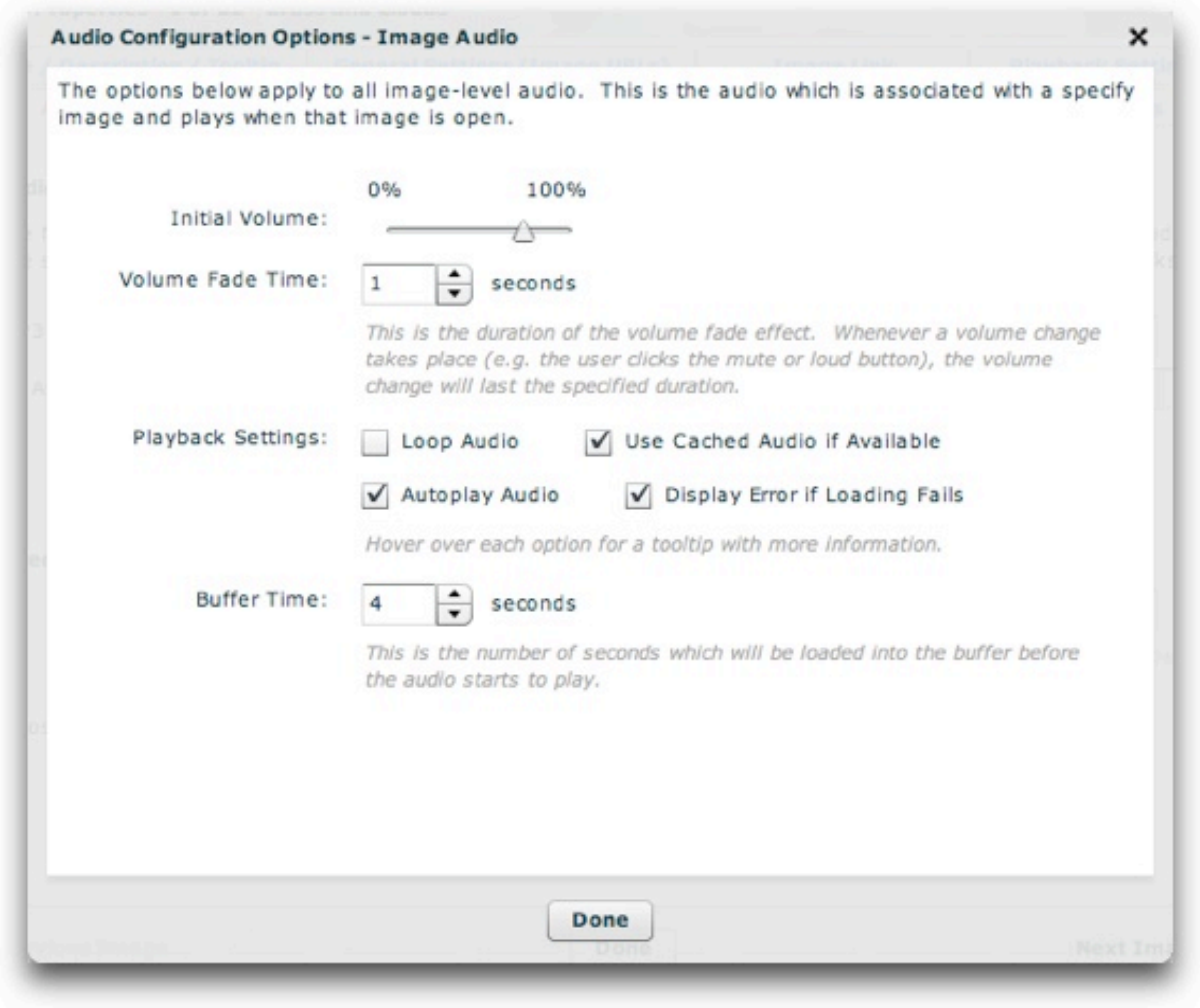
Once you specify an audio file, you can enter a string value in the Audio Label field. This value represents the label associated with the audio file; it is shown in any related audio controls while the file is playing. It should be the title of the song or the narration, for example.

Editing the Image-Level Audio Configuration Settings

There are several configuration settings, specific to image-level audio, which can be specified. Note that **these settings apply to all image-level audio**, not just a specific slide. These options are accessed through the 'Edit Image-Level Audio Configuration Settings...' button.



The dialog appears as follows. Note that the settings in this dialog apply to all image-level audio, not just the audio applied to a single image/slide.



Audio Configuration Options - Image Audio

The options below apply to all image-level audio. This is the audio which is associated with a specify image and plays when that image is open.

Initial Volume: 0% 100%

Volume Fade Time: 1 seconds

This is the duration of the volume fade effect. Whenever a volume change takes place (e.g. the user clicks the mute or loud button), the volume change will last the specified duration.

Playback Settings: □ Loop Audio ☑ Use Cached Audio if Available ☑ Autoplay Audio ☑ Display Error if Loading Fails

Hover over each option for a tooltip with more information.

Buffer Time: 4 seconds

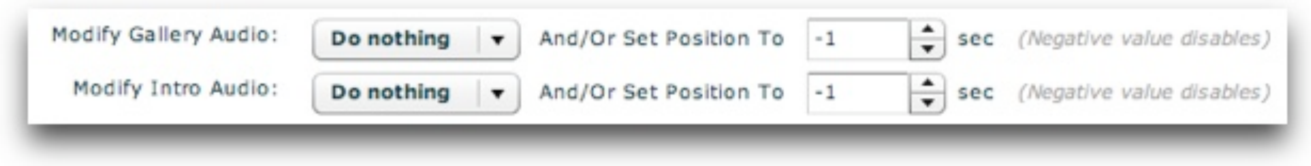
This is the number of seconds which will be loaded into the buffer before the audio starts to play.

Each of the options controls the following (when working in the wizard, you can also learn more about some settings directly by hovering over their names):

Property	Description
Initial Volume	This is the initial volume of the audio file when it begins playing, excepting any user adjustments via a volume control. If you have provided user controls for volume, then this is simply the starting point. A value of 0% is entirely silent, whereas 100% is full volume.
Volume Fade Time	The volume fade effect fades the volume from its current value to the new value over the duration specified by this setting. Especially if you are using mute/loud volume controls (which jump directly to a certain volume), it is recommended that you utilize this effect. This provides a polished, less jarring transition when muting and unmuting audio.

Loop Audio	Specifies if the audio should loop back to the beginning when it finishes playing.
Use Cached Audio if Available	Rarely should this setting be disabled. If enabled, the slideshow will use a cached version of the audio file from the user's internet cache when available. This will improve loading performance if the user listens to the same file more than once, making it start almost instantly on subsequent loads. This setting should only be disabled in cases where the audio file is frequently updated.
Autoplay Audio	Specifies if the audio should start playing automatically. If this option is disabled, you must give the user audio controls and they must manually play the audio.
Display Error if Loading Fails	By default, an error will be shown when the loading of the audio fails. In a production setup, you may want to disable this setting. In rare instances on certain configurations, the audio may play correcly yet an error message appears. In these cases, the problem can be avoided by disabling the error reporting.
Buffer Time	This is the number of seconds of audio which must be loaded into the buffer before the audio starts to play. Additionally, if the buffer runs out, this duration will be loaded before playback resumes. Generally the default value of about 4 seconds is a good setting; it will ensure that even on slow connections the audio doesn't cut in and out.

Controlling the Gallery and Intro Level Audio



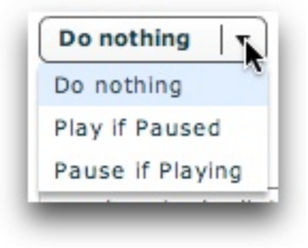
Modify Gallery Audio: Do nothing ▾ And/Or Set Position To -1 sec (Negative value disables)

Modify Intro Audio: Do nothing ▾ And/Or Set Position To -1 sec (Negative value disables)

The final two options in the Audio section allow you to control the gallery-level* and intro-level* audio. These settings allow you to either play or pause this audio, as well as set its position. Example uses include:

- Sync a gallery narration (or song) to specific timing points for each slide
- Ensure that background (intro-level) audio is paused while a single slide narration plays
- Make sure that audio is playing at a certain point
- Only play intro-level audio while certain galleries are open

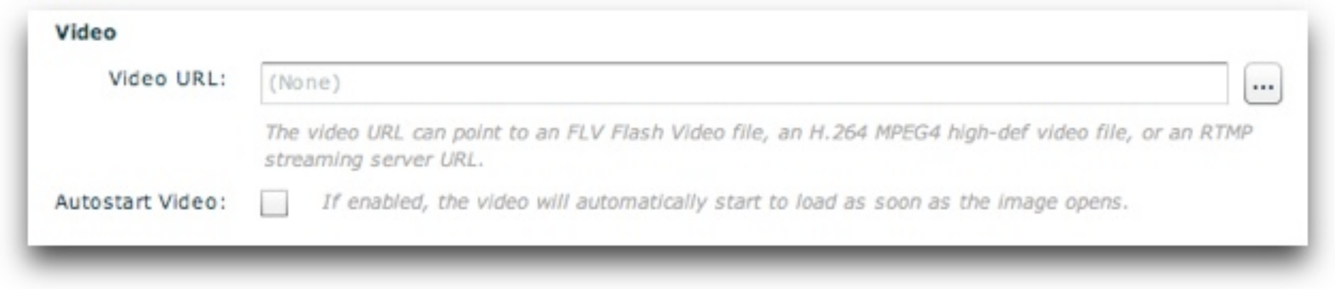
By default, none of these options are enabled. To make the gallery or intro audio be forced to play if paused or pause if playing, select one of the options from the dropdown:



To sync the gallery or intro audio to a certain timing point when the slide opens, enter a positive numeric value in the 'Set Position To' numeric stepper box. The value is denominated in seconds. Using a negative value (as is the default) disables any set-position jump; the position won't be modified in this case.

**Gallery-level audio is associated with a single gallery and only plays while it is open. Intro-level audio plays as a background for the entire slideshow, and is not associated with any specific image or gallery.*

Adding Video



Video

Video URL: (None) ...

The video URL can point to an FLV Flash Video file, an H.264 MPEG4 high-def video file, or an RTMP streaming server URL.

Autostart Video: □ If enabled, the video will automatically start to load as soon as the image opens.

This is the section where you can add a video file to this slide. The video URL can point to any of the following types of files:

- A standard FLV Flash Video File located within your site or hosted at some remote location (see cross-domain loading note below)
- An H.264 MPEG4 high-definition video file located within your site
- An RTMP streaming video URL (usually delivered through the Flash Media Server).

Enter the URL in the box or use the browse button to the right ('...') to select a file. **Very Important:** If using a relative URL, the URL must be relative to the supporting files folder, NOT the HTML page holding the slideshow. All other relative URLs in the slideshow (for images, audio, etc) are relative to the HTML page, whereas relative video URLs must be relative to the supporting files folder. Using the browse button ensures that this path is automatically created properly.

Note: If you are loading audio from a **different domain** than the one on which the slideshow is displayed, you will be affected by Flash Player crossdomain loading security restrictions. To learn more, see this topic.

Note for advanced users: To learn about how to customize the appearance of the video player, see the `com.dwuser.ss4.video.FlexVideoPlayer` class in the API Reference.

Once you have specified a video file, there are two options for when the video playback begins. If you leave the 'Autostart Video' checkbox *disabled*, then the user must click on the image for the video to start. If you use this option, you might want to add a tooltip value such as 'Click to play video.' If you *enable* the 'Autostart Video' checkbox, then the video will begin playing as soon as the image opens. If the video is configured to begin as soon as the slide opens, you can set an Autoplay Delay (see the Playback Settings tab) which tells the slideshow to automatically continue as soon as the video completes, or after a certain delay if video loading fails.

XML Equivalents (for Advanced Users and Developers)

The settings above map to the following XML attributes on each `<img ... />` node:

Property	Equivalent <img ... /> Attribute	Notes
MP3 Audio File	`@audio`	
Audio Label	`@audioLabel`	
Gallery Audio Play/Pause	`@galleryAudioPlay`	Not specifying an attribute is the same as 'Do Nothing'. Specifying `true` will force the audio to play; specifying `false` will force the audio to stop/pause.
Gallery Audio Set Position	`@galleryAudioPosition`	Denominated in milliseconds (whereas wizard is seconds).
Intro Audio Play/Pause	`@introAudioPlay`	Not specifying an attribute is the same as 'Do Nothing'. Specifying `true` will force the audio to play; specifying `false` will force the audio to stop/pause.
Intro Audio Set Position	`@introAudioPosition`	Denominated in milliseconds (whereas wizard is seconds).
Video URL	`@video`	
Autostart Video	`@videoAutoStart`	Supported values are `true` and `false`.

Note that the '@' is part of customary syntactical representation, and is not part of each attribute name.

The Image-Level Audio Configuration Settings map to the `<AudioImage ... />` styles node within the `<styles>` section of the XML configuration file. The settings in the above table map to the following attributes on that node:

Property	Equivalent <AudioImage ... /> Attribute	Notes
Initial Volume	`@volume`	Supported values range from 0 to 100.
Volume Fade Time	`@volumeChangeTime`	Denominated in milliseconds (whereas wizard is seconds).
Loop Audio	`@loop`	Supported values are `true` and `false`.
Use Cached Audio if Available	`@useCache`	Supported values are `true` and `false`.
Autoplay Audio	`@autoplay`	Supported values are `true` and `false`.
Display Error if Loading Fails	`@errorReporting`	Supported values are `true` and `false`.
Buffer Time	`@buffer`	Denominated in seconds (same as wizard).

Note that the '@' is part of customary syntactical representation, and is not part of each attribute name.

For more detailed XML information, see the Developer Reference.

Panning / Zooming Effects (Ken Burns Effects)

The Ken Burns Effects tab is where you can apply Ken Burns (KB) effects, which incorporate both panning and zooming, to your image/slide. The tab appears as follows *(note -- in the screenshots below, the tab is labeled 'Panning / Zooming Effects', whereas currently the wizard displays 'Ken Burns Effects')*:

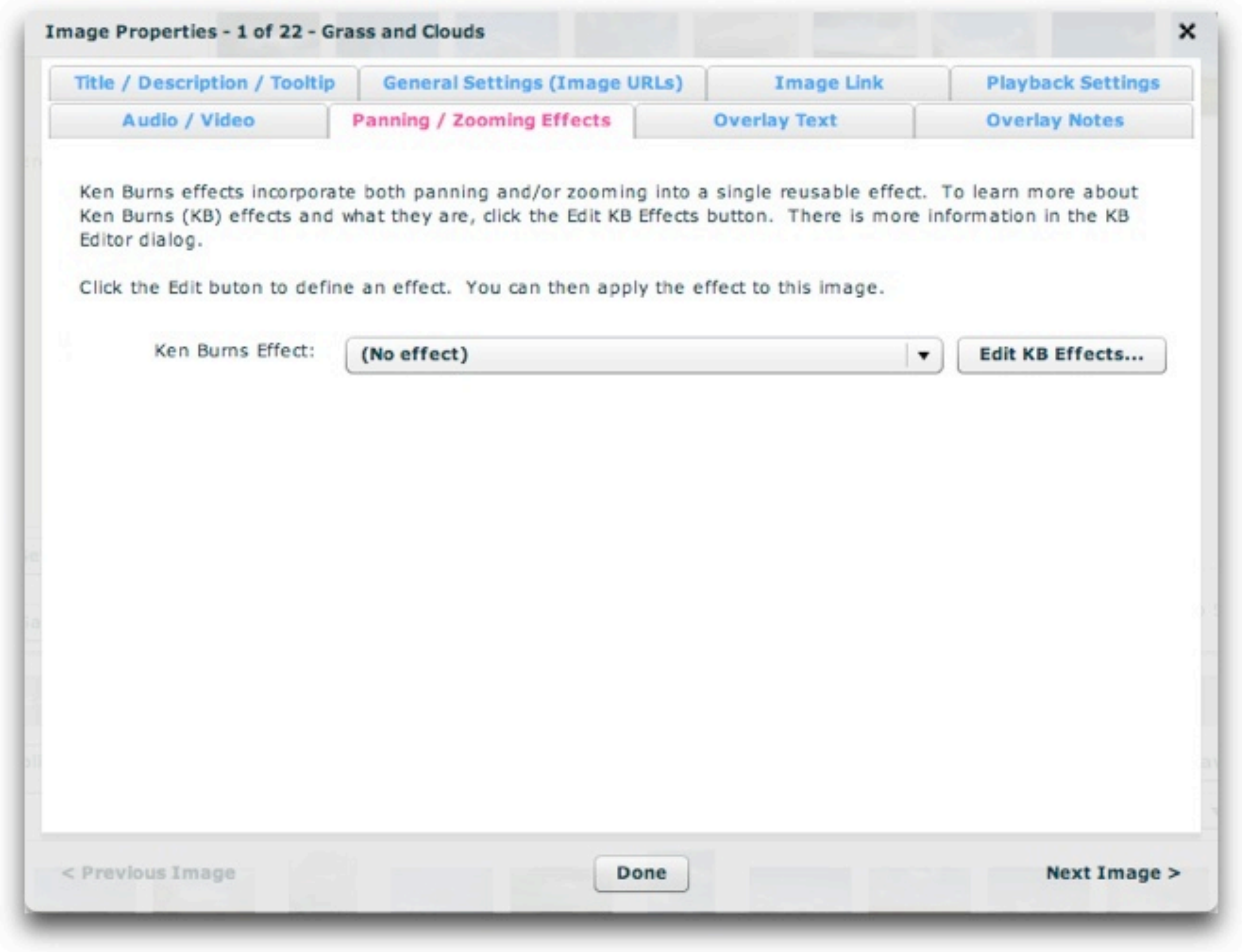


Image Properties - 1 of 22 - Grass and Clouds
Title / Description / Tooltip
General Settings (Image URLs)
Image Link
Playback Settings
Audio / Video
Panning / Zooming Effects
Overlay Text
Overlay Notes
Edit KB Effects...
< Previous Image
Done
Next Image >

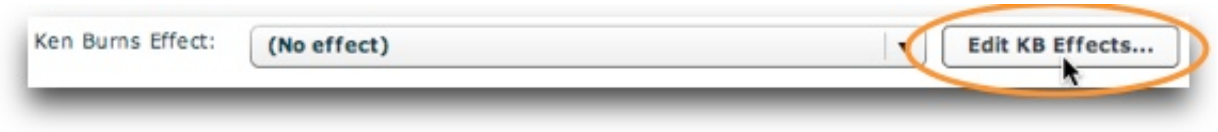
Ken Burns effects incorporate both panning and/or zooming into a single reusable effect. To learn more about Ken Burns (KB) effects and what they are, click the Edit KB Effects button. There is more information in the KB Editor dialog.

Click the Edit buton to define an effect. You can then apply the effect to this image.

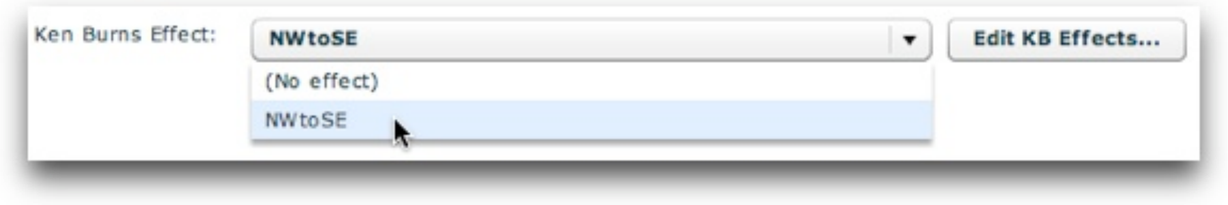
Ken Burns Effect: (No effect)

Ken Burns effects (KB effects) are packages which incorporate both panning and zooming. You can learn more about KB effects in the topic on using the KB Effect Editor (there is a link to this topic below). Any value specified here is image-specific and overrides any gallery default established in the Gallery Properties dialog. The default value, **(Inherit Gallery-wide Default)**, means that the gallery-wide default will be used.

To be able to select an effect, you must first create (define) effects using the KB Effect Editor. This dialog is accessed by clicking the 'Edit KB Effects...' button:


Ken Burns Effect: (No effect) Edit KB Effects...

This will launch the KB Effect Editor. To learn about how to use this window to define effects, see Using The Ken Burns Effect (Panning / Zooming) Editor. After creating an effect and returning to the Image Properties dialog, you'll be able to select the effect you just created from the dropdown list:


Ken Burns Effect: NWtoSE Edit KB Effects...

After defining multiple KB effects, you can also choose the **(Random Effect)** option from the dropdown (this option is not shown in the above screenshot). When this option is selected, an effect will randomly be selected from among all defined effects each time the image / slide is displayed. Or, you can also choose the **(Cycle All Effects)** option from the dropdown (this option is not shown in the above screenshot). When this option is selected, all created effects will be cycled through as KB effects are displayed.

XML Equivalents (for Advanced Users and Developers)

The settings above map to the following XML attributes on each `<img ... />` node:

Property	Equivalent <img ... /> Attribute	Notes
Ken Burns Effect	`@kbEffect`	In the XML, you can specify multiple KB effect ID values; when doing this, separate each ID with a comma (no spaces). One of these multiple effects will be randomly chosen each time an effect is to be applied. Omitting the attribute causes the gallery-wide default to be inherited. Using a value of `None` will override any existing gallery-wide default, disabling the KB effect for that image / slide. Using a value of `Random` causes all defined KB effects to randomly be used. Using a value of `Cycle` causes all defined KB effects to be cycled through.

Note that the '@' is part of customary syntactical representation, and is not part of each attribute name.

For more detailed XML information, see the Developer Reference.

Overlay Text

The Overlay Text tab is where you can specify text which is overlaid over the image, in a manner similar to Microsoft Powerpoint. Each separate box of overlay text can be timed to appear and disappear at a specific point with a customized effect. The text styling is fully customizable as well. HTML can be included in the text for event more control. The tab appears as follows:

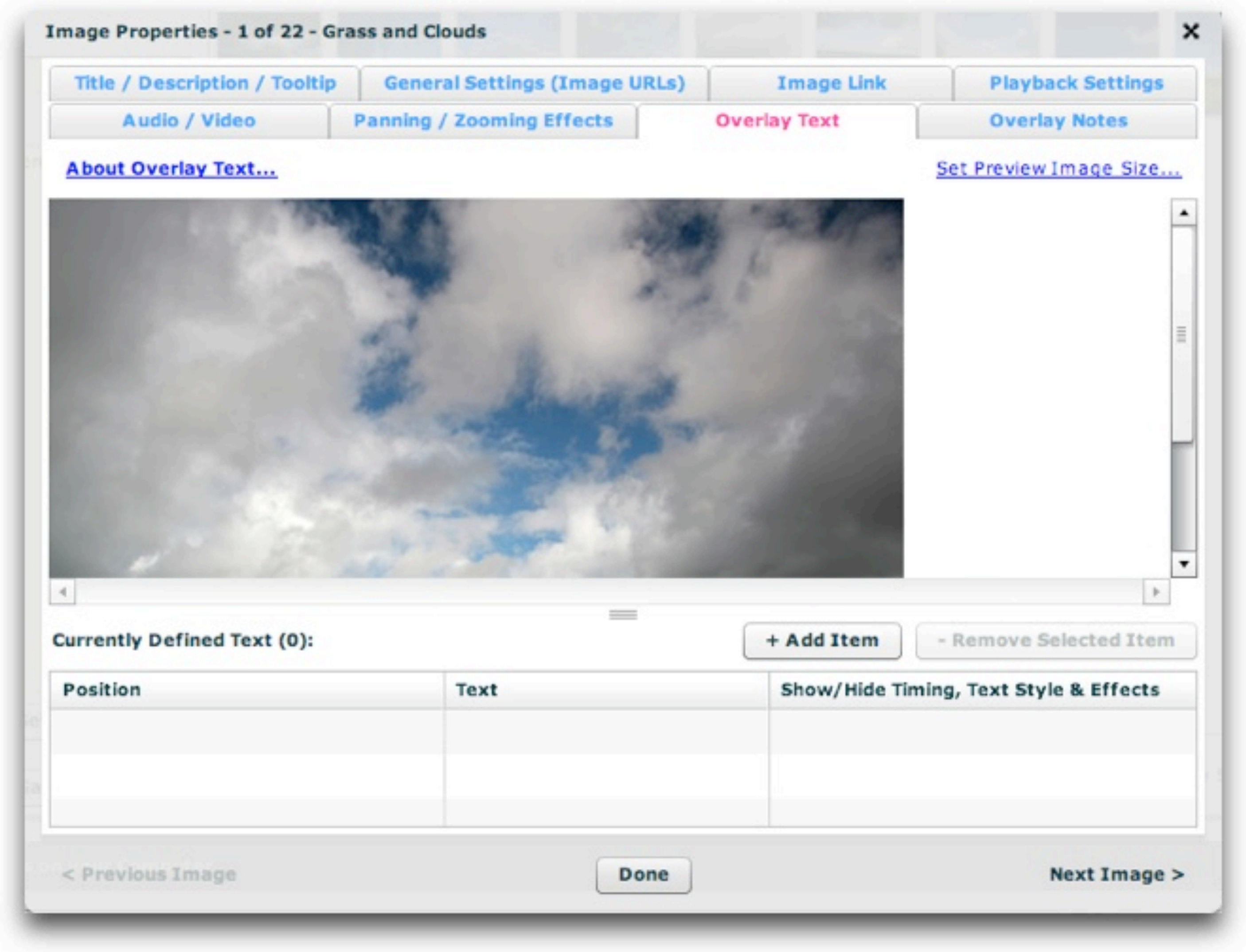
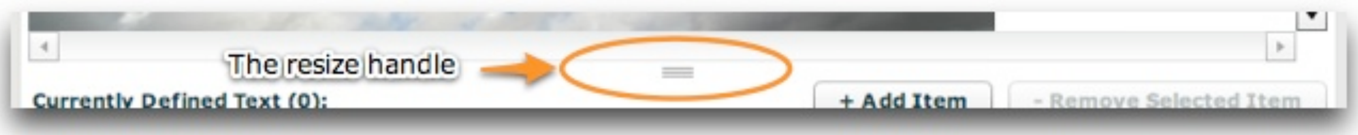
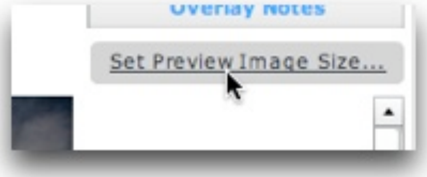


Image Properties - 1 of 22 - Grass and Clouds
Title / Description / Tooltip
General Settings (Image URLs)
Image Link
Playback Settings
Audio / Video
Panning / Zooming Effects
Overlay Text
Overlay Notes
About Overlay Text...
Set Preview Image Size...
Currently Defined Text (0):
+ Add Item
- Remove Selected Item
Position
Text
Show/Hide Timing, Text Style & Effects
< Previous Image
Done
Next Image >

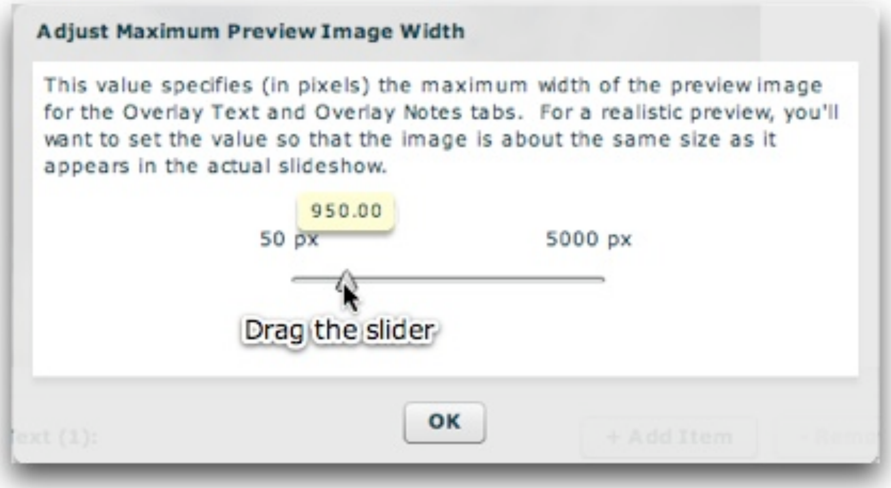
On the top half of the window, there is the preview image. This allows you to position the text over the image while seeing how it will appear when the slideshow is viewed. On the bottom half of the window, a list of all overlay text on the slide is displayed. To resize either of these parts of the window, use the resize handle between them:



By default, the preview image is shown at its full size. In some instances, this may not be a realistic preview because your slideshow will display the image at a scaled-down size. In order to make the preview more realistic, you can change the maximum width of the preview image, thereby forcing it to resize down. To do this, click the 'Set Preview Image Size...' button at the top right:



In the Set Preview Size dialog which appears, you can drag the slider to set the new maximum width of the image:

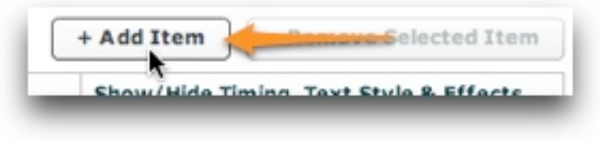

Adjust Maximum Preview Image Width

This value specifies (in pixels) the maximum width of the preview image for the Overlay Text and Overlay Notes tabs. For a realistic preview, you'll want to set the value so that the image is about the same size as it appears in the actual slideshow.

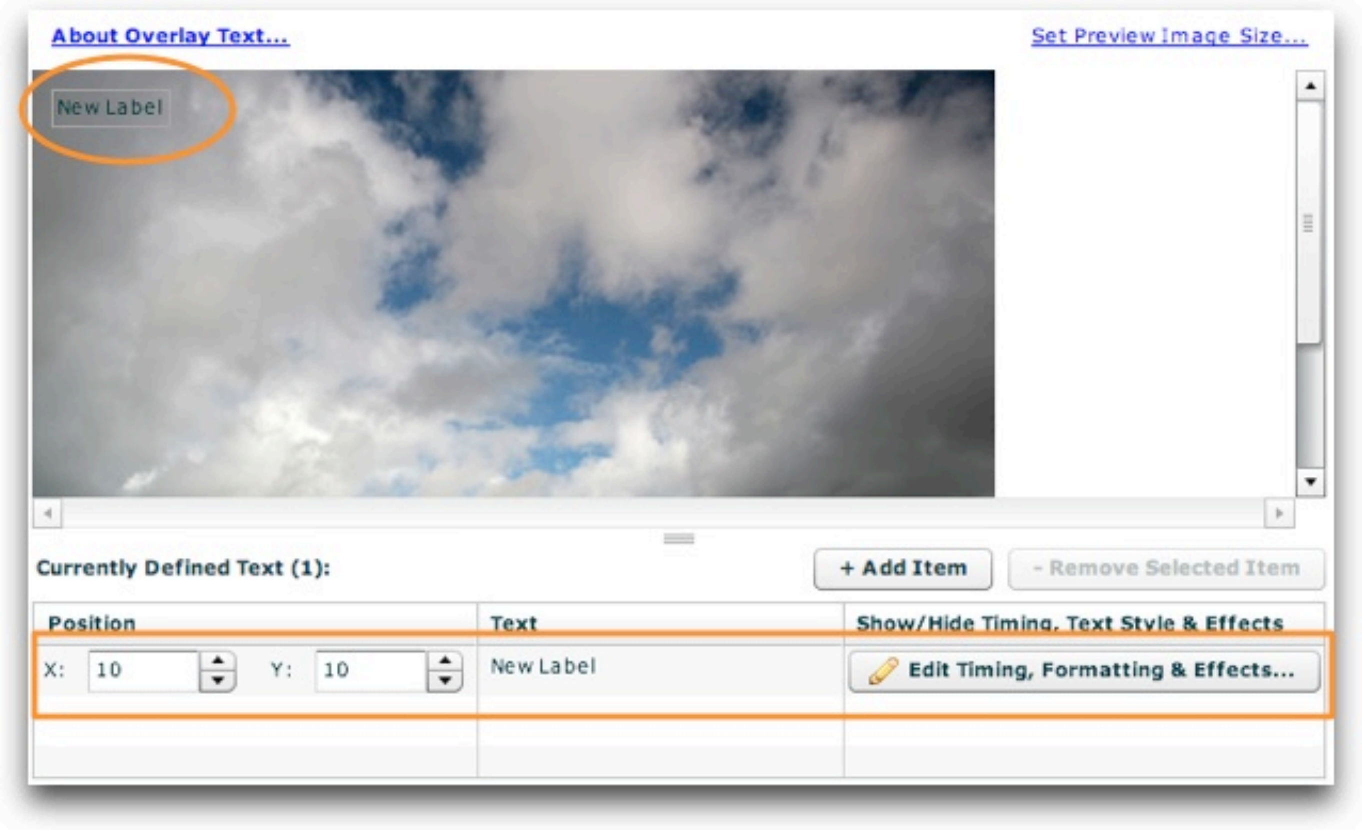
The default value is 5000 pixels, which ensures that the image isn't scaled at all. A smaller value will force resizing to occur. After making any changes, click the OK button to return the main dialog.

Adding and Configuring Overlay Text

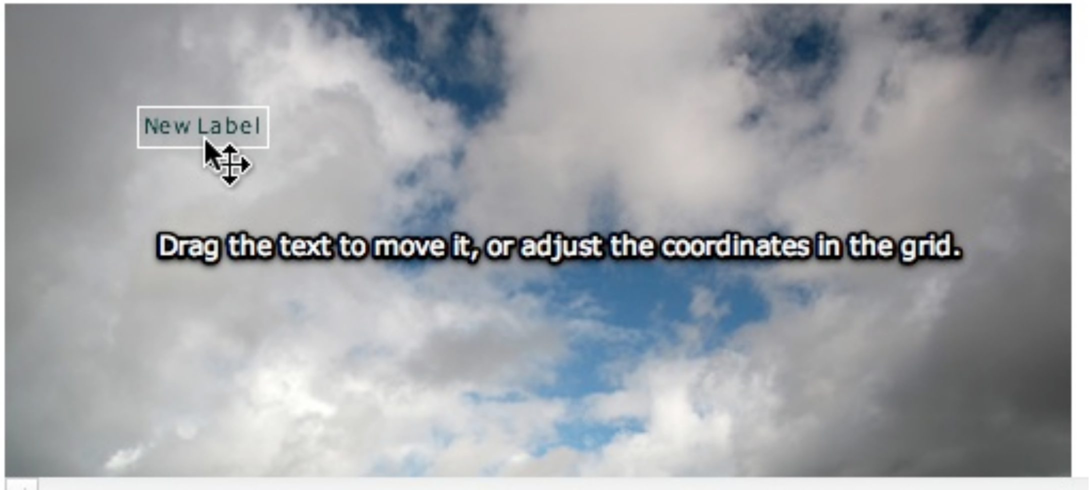
To add overlay text, click the '+ Add Item' button:



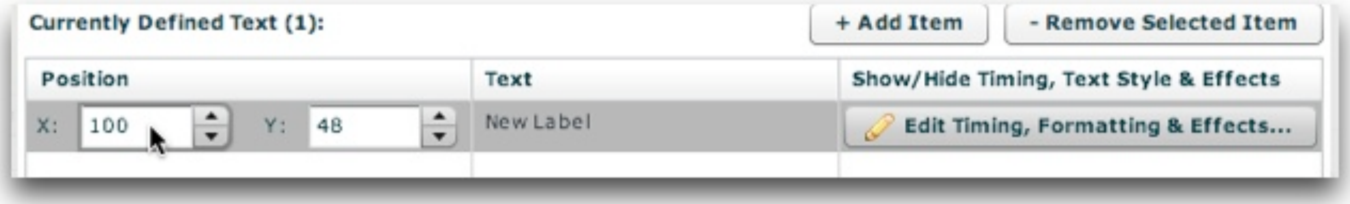
A new overlay text box will appear on the preview image, and a corresponding item will be added to the grid at the bottom:


About Overlay Text...
Set Preview Image Size...
New Label
Currently Defined Text (1):
+ Add Item
- Remove Selected Item
Position
Text
Show/Hide Timing, Text Style & Effects
X: 10
Y: 10
New Label
Edit Timing, Formatting & Effects...

You can reposition the text in one of two different ways. First, you can drag the label around on top of the preview image:

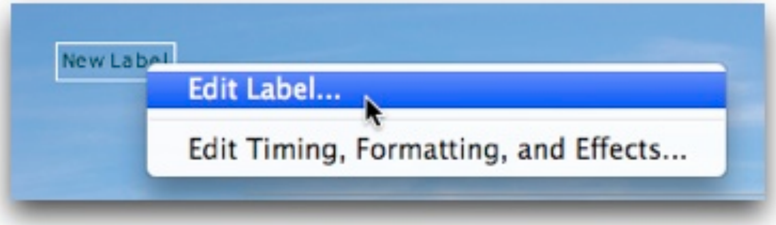

New Label
Drag the text to move it, or adjust the coordinates in the grid.

Or, you can set the X and Y coordinates directly in the grid:

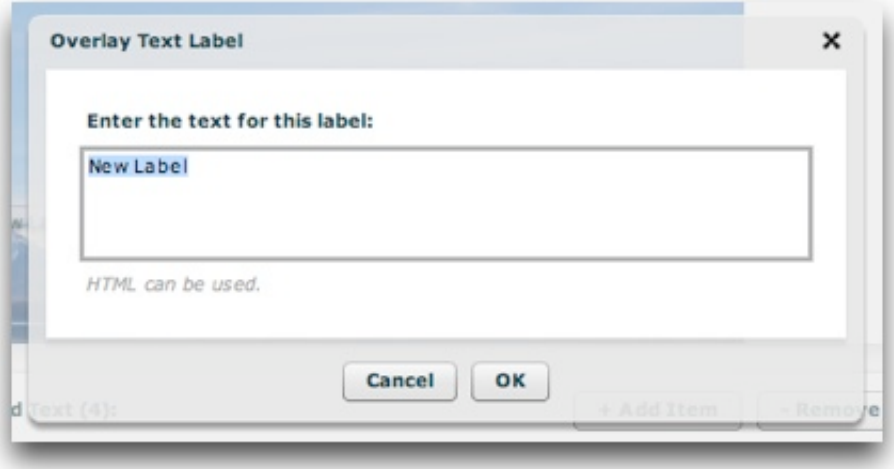

Currently Defined Text (1):
+ Add Item
- Remove Selected Item
Edit Timing, Formatting & Effects...

Position	Text	Show/Hide Timing, Text Style & Effects
X: 100 Y: 48	New Label	Edit Timing, Formatting & Effects...

To set the text, double-click the preview label, or Right/CMD-click it and select 'Edit Label':



This will open a dialog where you can change the text (HTML can be used; multiline text is supported as well), then click OK:



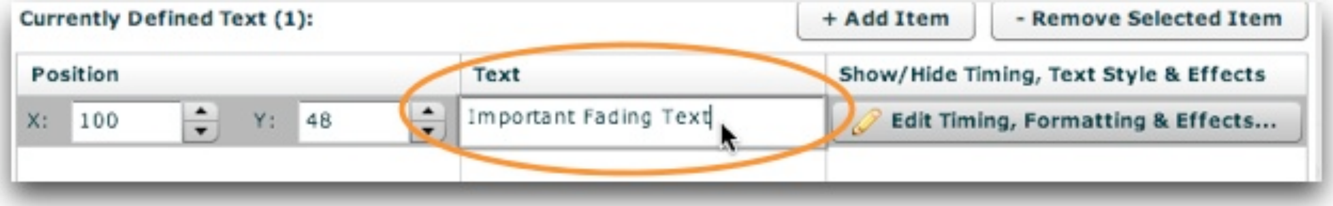
Overlay Text Label

Enter the text for this label:

New Label

HTML can be used.

You can also set the text by clicking the label value in the grid. It will become an editable textbox where you can enter the desired value:

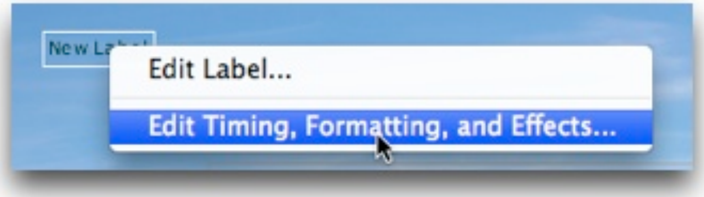


Currently Defined Text (1):

Position	Text	Show/Hide Timing, Text Style & Effects
X: 100 Y: 48	Important Fading Text	Edit Timing, Formatting & Effects...

As noted earlier, HTML can be used within the text value. After changing the text, the preview text which floats over the preview image will be updated as well.

To edit the timing (when the text appears and hides), formatting (how the text looks) and effects (used for showing and hiding), Right/CMD-click the label and select 'Edit Timing, Formatting, and Effects...':

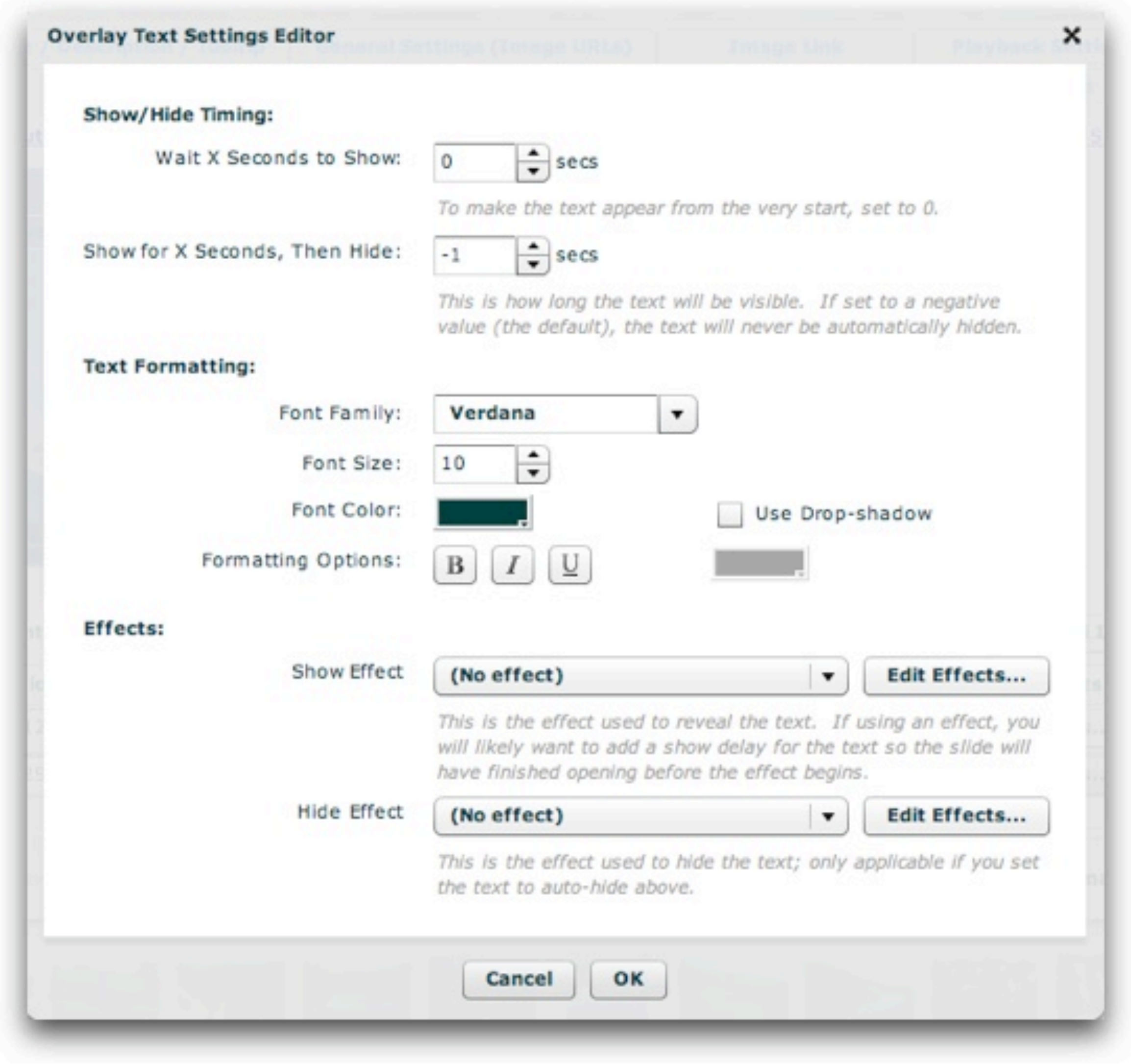


Note: You can also access these settings by clicking the 'Edit Timing, Formatting & Effects...' button in the grid:

Selecting either of these options will launch the Overlay Text Settings Editor dialog.

Setting Text Properties in the Overlay Text Settings Dialog

Accessed by clicking the edit button in the overlay text grid as mentioned above, the Overlay Text Settings Editor dialog appears as follows:



Overlay Text Settings Editor

Show/Hide Timing:

Wait X Seconds to Show: 0 secs

To make the text appear from the very start, set to 0.

Show for X Seconds, Then Hide: -1 secs

This is how long the text will be visible. If set to a negative value (the default), the text will never be automatically hidden.

Text Formatting:

Font Family: Verdana

Font Size: 10

Font Color:

□ Use Drop-shadow

Formatting Options: B I U

Effects:

Show Effect (No effect) Edit Effects...

This is the effect used to reveal the text. If using an effect, you will likely want to add a show delay for the text so the slide will have finished opening before the effect begins.

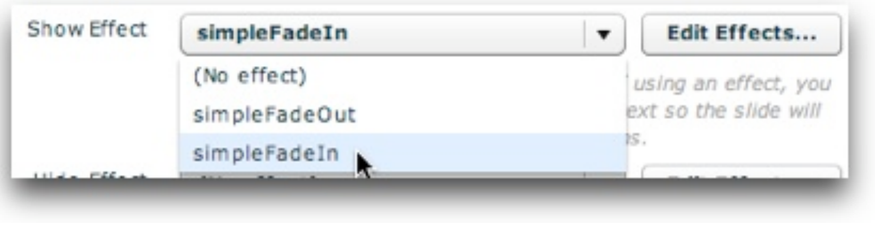
Hide Effect (No effect) Edit Effects...

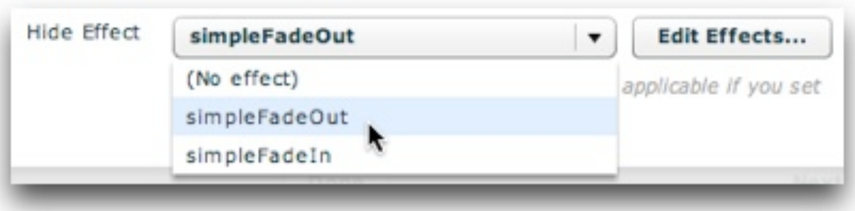
This is the effect used to hide the text; only applicable if you set the text to auto-hide above.

Cancel OK

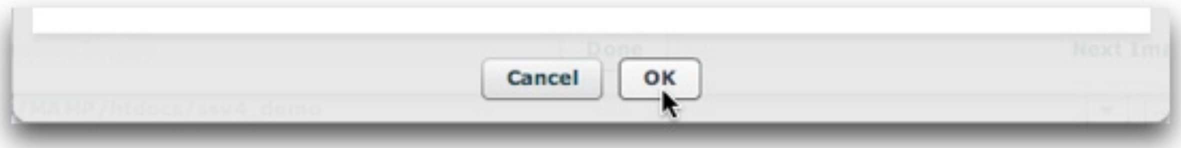
Each option controls the following. If any option is not specified, the default value is used.

Property	Description
Wait for X Seconds to Show	This value can be used to specify a delay before the text appears on the image. The default value, 0, means that the text will be visible as soon as the slide opens.

(AKA Show-delay)	If you use a Show Effect, you'll likely want to add a small show-delay for this option so that the slide will have fully opened before the text appears. *For an example, see the Timing Examples below.*
Show for X Seconds, Then Hide *(AKA Visible-time)*	This value can be used to make the text automatically disappear after a certain length of time. If set to a negative value (the default), the text will never be automatically hidden; it will remain visible until the slide is closed or removed. *For an example, see the Timing Examples below.*
Font Family	This is the font face used on the text. Select from the dropdown list, which includes all universally web-safe fonts.
Font Size	Specifies the size of the text in points.
Font Color	Specifies the color of the text.
Formatting Options	Each button toggles a text styling option -- Bold, Italic, and Underline in that order.
Use Drop-Shadow	If enabled, a drop shadow will be added to the text. The color of the shadow is controlled by the color picker directly below the checkbox.
Drop-Shadow Color	If the Use Drop-shadow option is enabled, this settings specifies the color of the shadow. The default value is black (#000000).
Show Effect	Specifies the effect used when the text is initially shown. If you specify a show-effect, you likely need to set a small show-delay so that the slide will be fully visible prior to the effect beginning. Failing to do so can make the effect not ever show. To select an effect, you must have already defined an effect using the Effect Editor, which can be accessed by clicking the 'Edit Effects...' button. To learn more about creating effects, see the Using the Effect Editor topic. After effects have been defined, you can select from them using the dropdown menu:  Show Effect simpleFadeIn Edit Effects... (No effect) simpleFadeOut simpleFadeIn using an effect, you ext so the slide will is.

Hide Effect	Specifies the effect used when the text is hidden, if a specific visible-time value has been set. If a negative visible-time value has been specified (indicating that the text never will be automatically hidden), then this parameter becomes irrelevant. To select an effect, you must have already defined an effect using the Effect Editor, which can be accessed by clicking the 'Edit Effects...' button. To learn more about creating effects, see the Using the Effect Editor topic. After effects have been defined, you can select from them using the dropdown menu:  Hide Effect simpleFadeOut Edit Effects... (No effect) simpleFadeOut simpleFadeIn applicable if you set

When you're finished making changes in the Overlay Text Settings Editor, click the OK button to save your changes and return to the main dialog, or click Cancel to discard your changes:



Timing Examples:

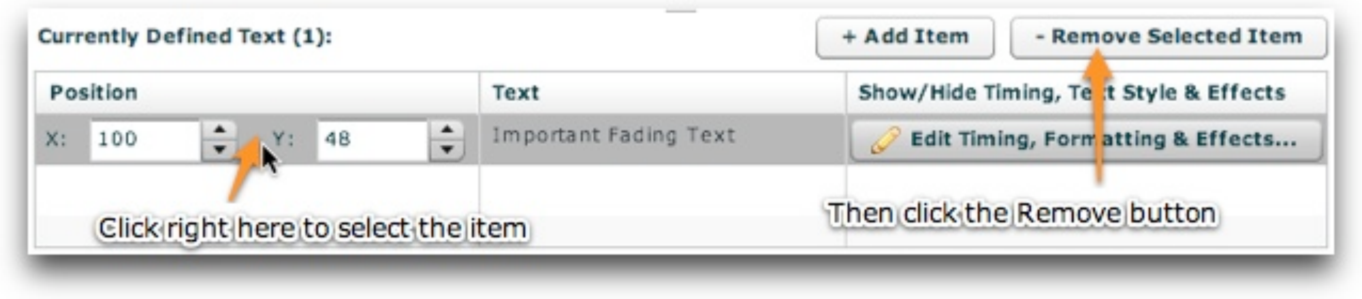
Suppose you added two overlay text boxes to your image - "Important Message 1" and "Important Message 2". If you wanted Important Message 1 to appear for a couple of seconds then disappear, then have Important Message 2 appear for a couple of seconds and disappear, you might use the following timing values:

Important Message 1: Show-delay = 1 sec; Visible-time = 4 sec.

Important Message 2: Show-delay = 6 sec; Visible-time = 4 sec.

Removing Overlay Text

To remove a textbox, you must first select the row. To do this, click just to the left of the 'Y' position box.   You can then click the '- Remove Selected Item' button:


Currently Defined Text (1):
+ Add Item
- Remove Selected Item
Position
Text
Show/Hide Timing, Text Style & Effects
X: 100
Y: 48
Important Fading Text
Edit Timing, Formatting & Effects...
Click right here to select the item
Then click the Remove button

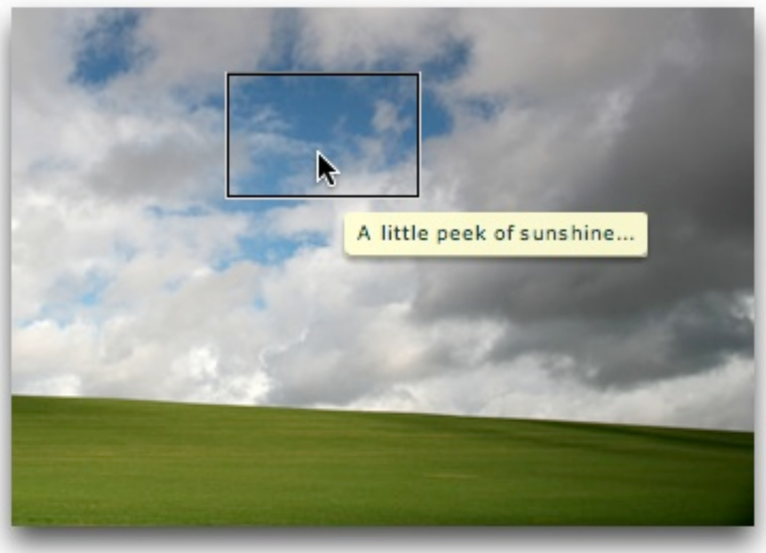
Advanced Customizations

This section is for advanced users or developers.

All overlay text instances are instances of the `mx.controls.Text` component.   As such, the `<text ... />` nodes in the `<img>` nodes which define the overlay text-boxes are actually `Text` instances.   They support most of the attributes available for the `Text` component; see the class reference for available parameters.

Overlay Notes

The Overlay Notes tab is where you can specify floating note boxes. These rectangular boxes, which are positioned over the image/slide, provide an information tooltip about that part of the image when hovered over. This allows you to add notes about specific features of a photo. HTML can be used within the tooltips. Generally, the boxes are only visible when the user hovers over the image, but this can be changed (see the Advanced Customizations section below). Here is an example note:



When the user hovers over the slide, the box becomes visible. When the user hovers over the box, the information tooltip becomes visible.

Back in the wizard, the tab appears as follows:

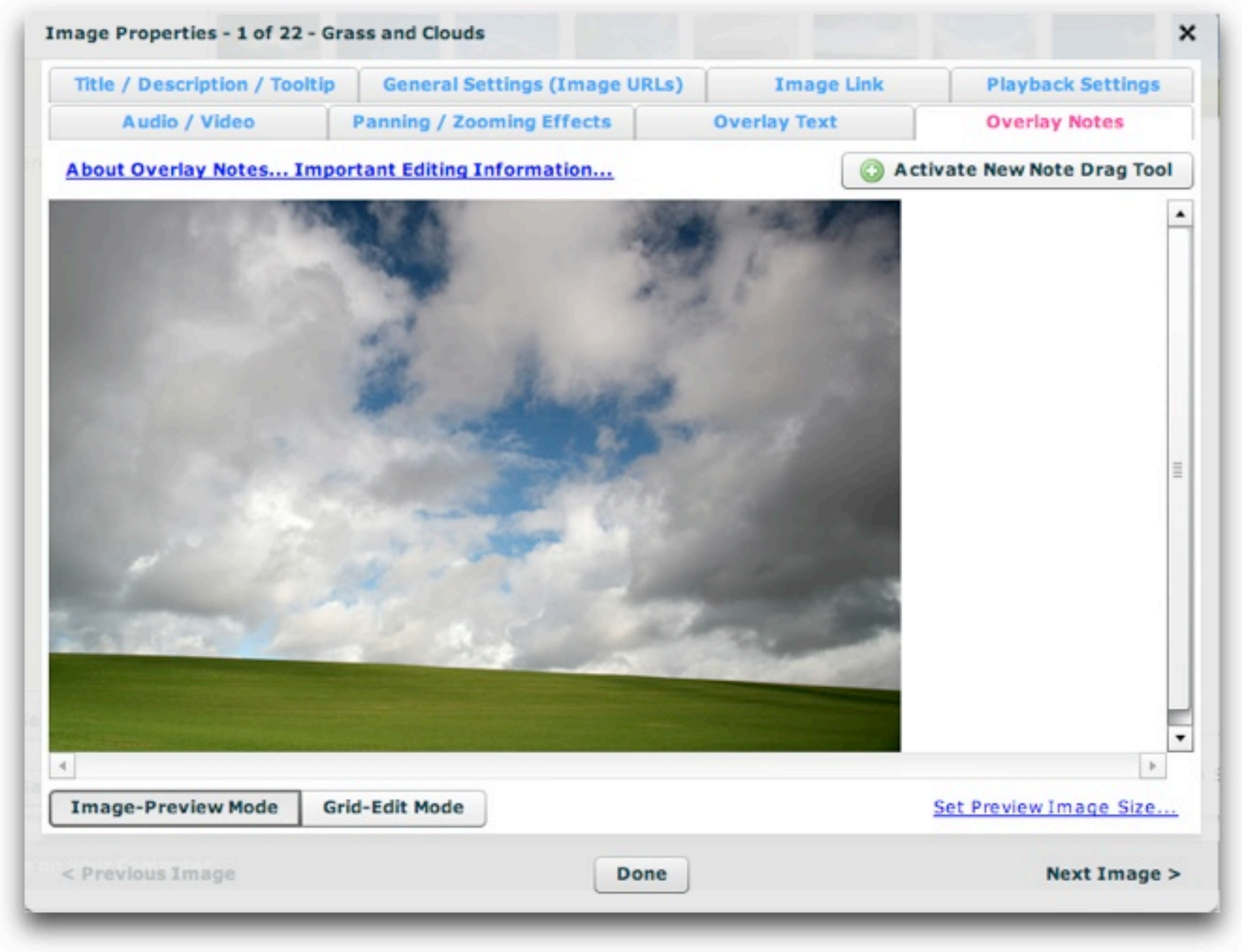
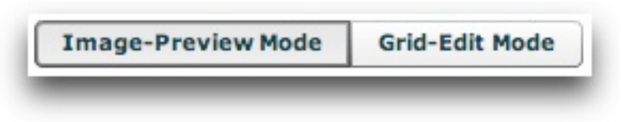
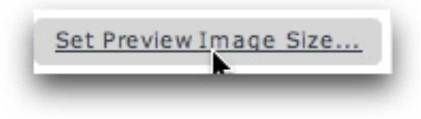


Image Properties - 1 of 22 - Grass and Clouds
Title / Description / Tooltip
General Settings (Image URLs)
Image Link
Playback Settings
Audio / Video
Panning / Zooming Effects
Overlay Text
Overlay Notes
About Overlay Notes... Important Editing Information...
Activate New Note Drag Tool
Image-Preview Mode
Grid-Edit Mode
Set Preview Image Size...
< Previous Image
Done
Next Image >

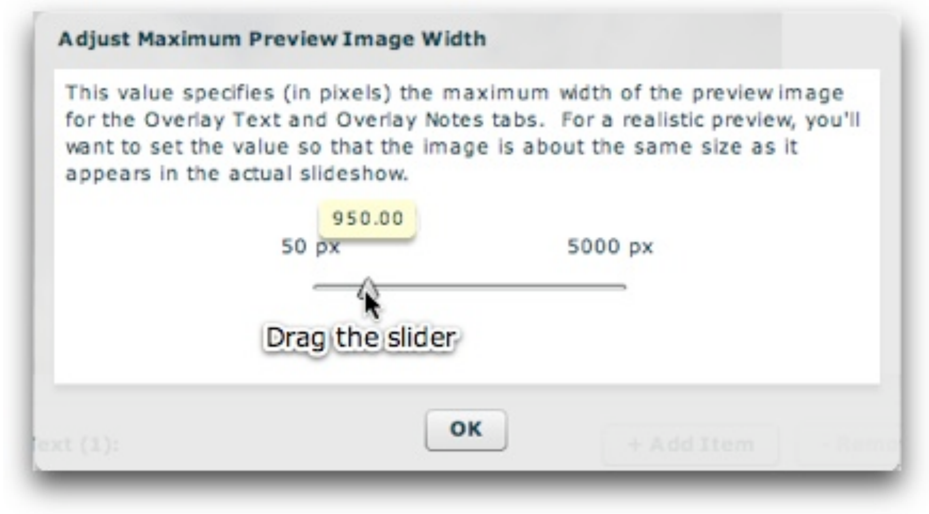
There are two views: the **Image-Preview** view, and the **Grid-Edit** view. The Image-Preview Mode allows you to position the note boxes over the image while seeing how it will appear when the slideshow is viewed. In Grid-Edit Mode, a grid listing of all overlay notes on the slide is displayed, allowing you to make changes to, add, or remove note boxes. These views are toggled at the bottom left by using these buttons:



By default, the preview image is shown at its full size. In some instances, this may not be a realistic preview because your slideshow will display the image at a scaled-down size. In order to make the preview more realistic, you can change the maximum width of the preview image, thereby forcing it to resize down. To do this, click the 'Set Preview Image Size...' button at the bottom right:



In the Set Preview Size dialog which appears, you can drag the slider to set the new maximum width of the image:



Adjust Maximum Preview Image Width

This value specifies (in pixels) the maximum width of the preview image for the Overlay Text and Overlay Notes tabs. For a realistic preview, you'll want to set the value so that the image is about the same size as it appears in the actual slideshow.

The default value is 5000 pixels, which ensures that the image isn't scaled at all. A smaller value will force resizing to occur. After making any changes, click the OK button to return the main dialog.

The Grid-Edit Mode appears as follows:

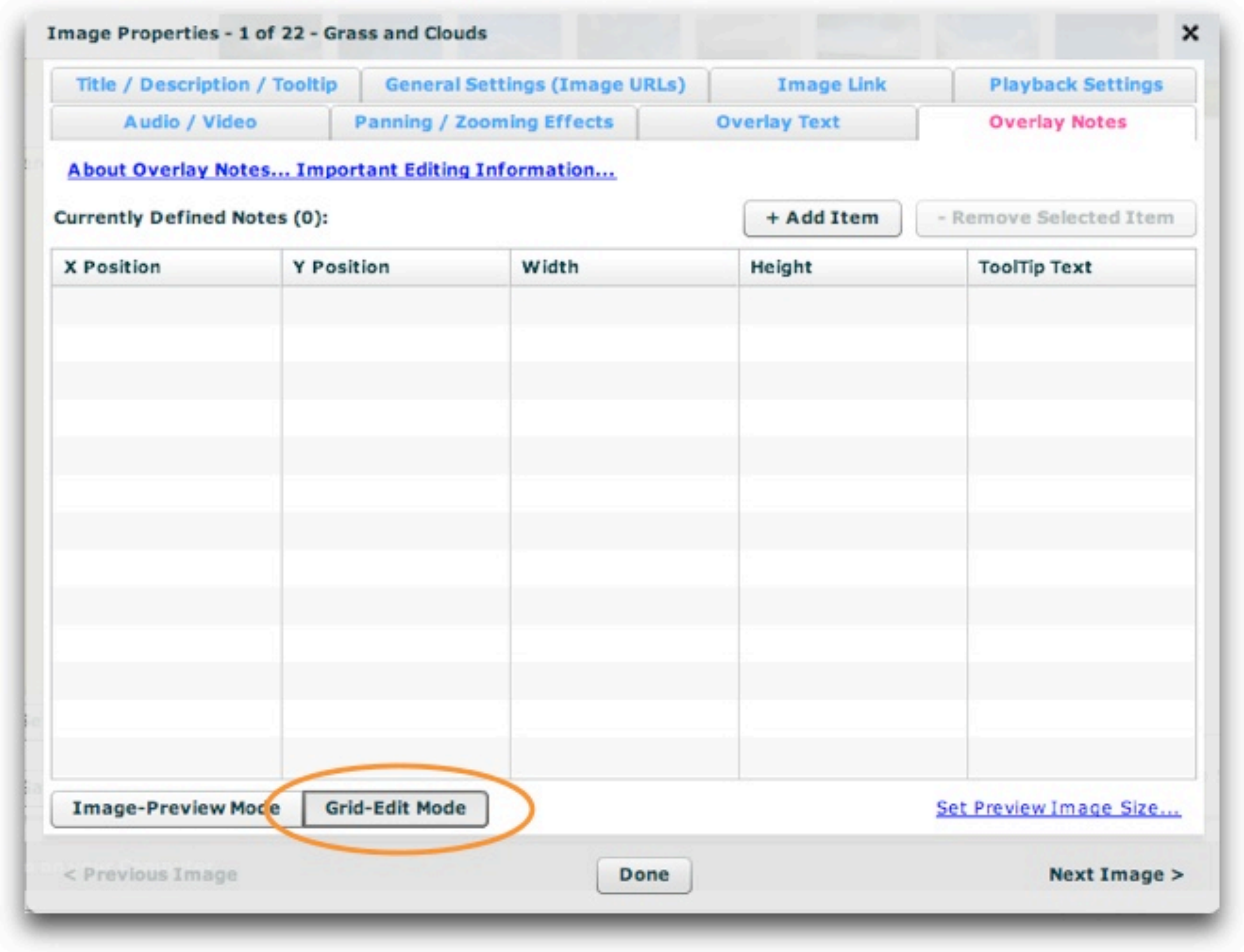


Image Properties - 1 of 22 - Grass and Clouds

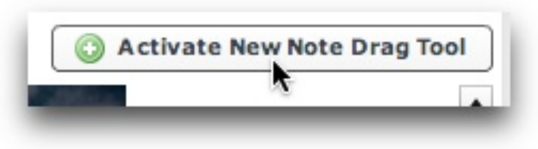
✕
Title / Description / Tooltip
General Settings (Image URLs)
Image Link
Playback Settings
Audio / Video
Panning / Zooming Effects
Overlay Text
Overlay Notes
About Overlay Notes... Important Editing Information...
+ Add Item
- Remove Selected Item
Image-Preview Mode
Grid-Edit Mode
Set Preview Image Size...
< Previous Image
Done
Next Image >

Currently Defined Notes (0):

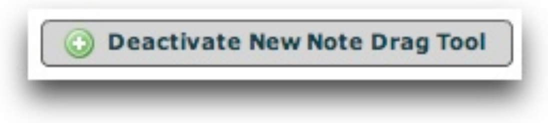
X Position	Y Position	Width	Height	ToolTip Text

Visually Adding/Modifying Notes using Image-Preview Mode

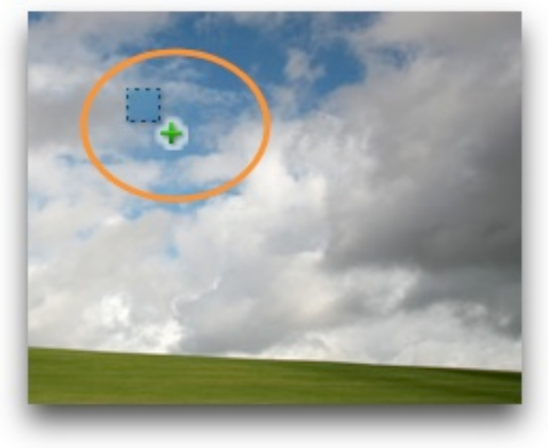
To add a note in the Image-Preview Mode, first activate the new-note drag tool by clicking the 'Activate New Note Drag Tool' button:



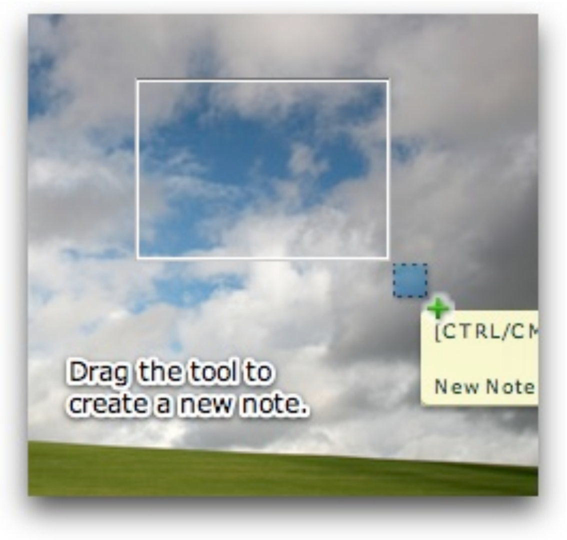
Note - If you want to deactivate the tool without adding a note, click the button again:



Once you have activated the tool, your cursor will change and appear as follows:

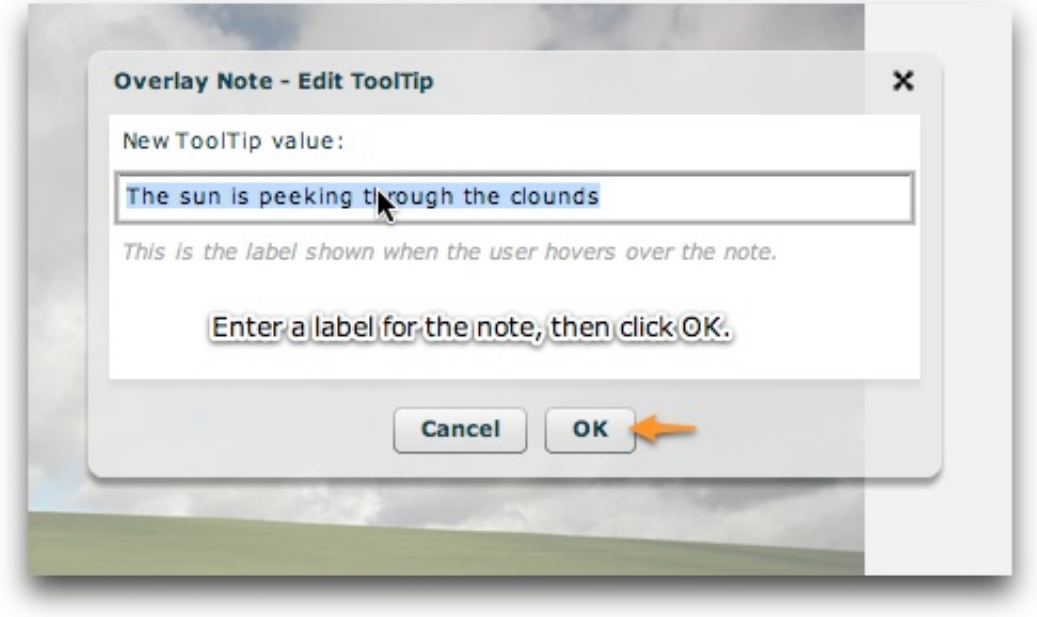


The registration point is the upper-left corner of the square with the dashed edge. To create a note, drag the tool across a portion of the image. It will look like this:



Note: A tooltip may appear while you're dragging. You can ignore it; it's not applicable when creating a note.

Once you release the mouse, you will be prompted to enter the tooltip text associated with the note. Note that you can enter HTML in the field.



Overlay Note - Edit ToolTip

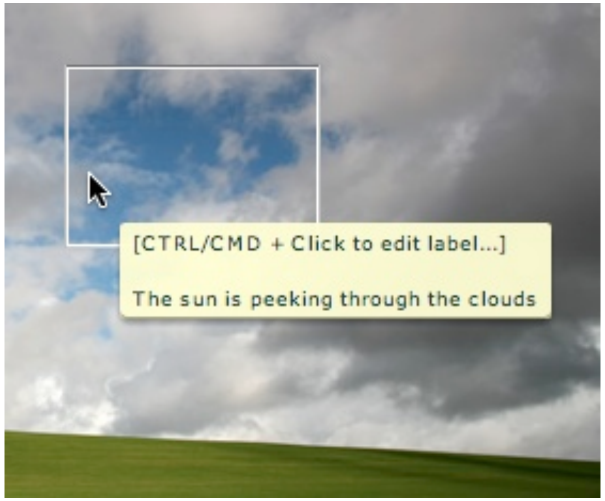
New ToolTip value:

The sun is peeking through the clounds

This is the label shown when the user hovers over the note.

Enter a label for the note, then click OK.

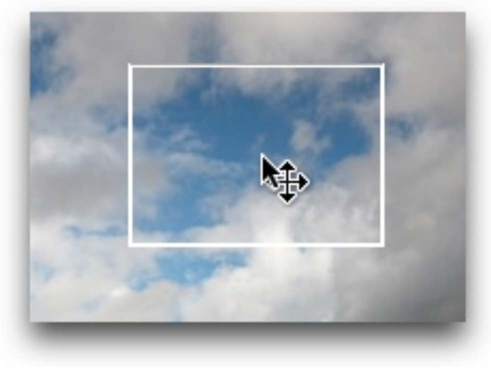
After entering the name, click OK. The new note will appear over the preview image. To see the tooltip associated with the label, hover over the box. To edit the label, CMD+Click (CTRL+Click Windows) on the box:



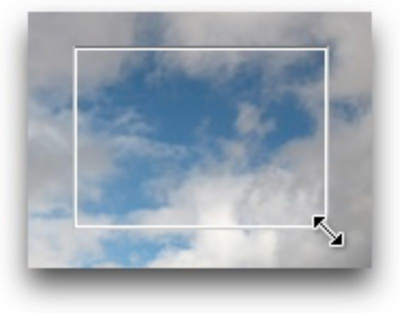
[CTRL/CMD + Click to edit label...]

The sun is peeking through the clouds

To reposition a note box, simply click in the middle of the box and drag it:

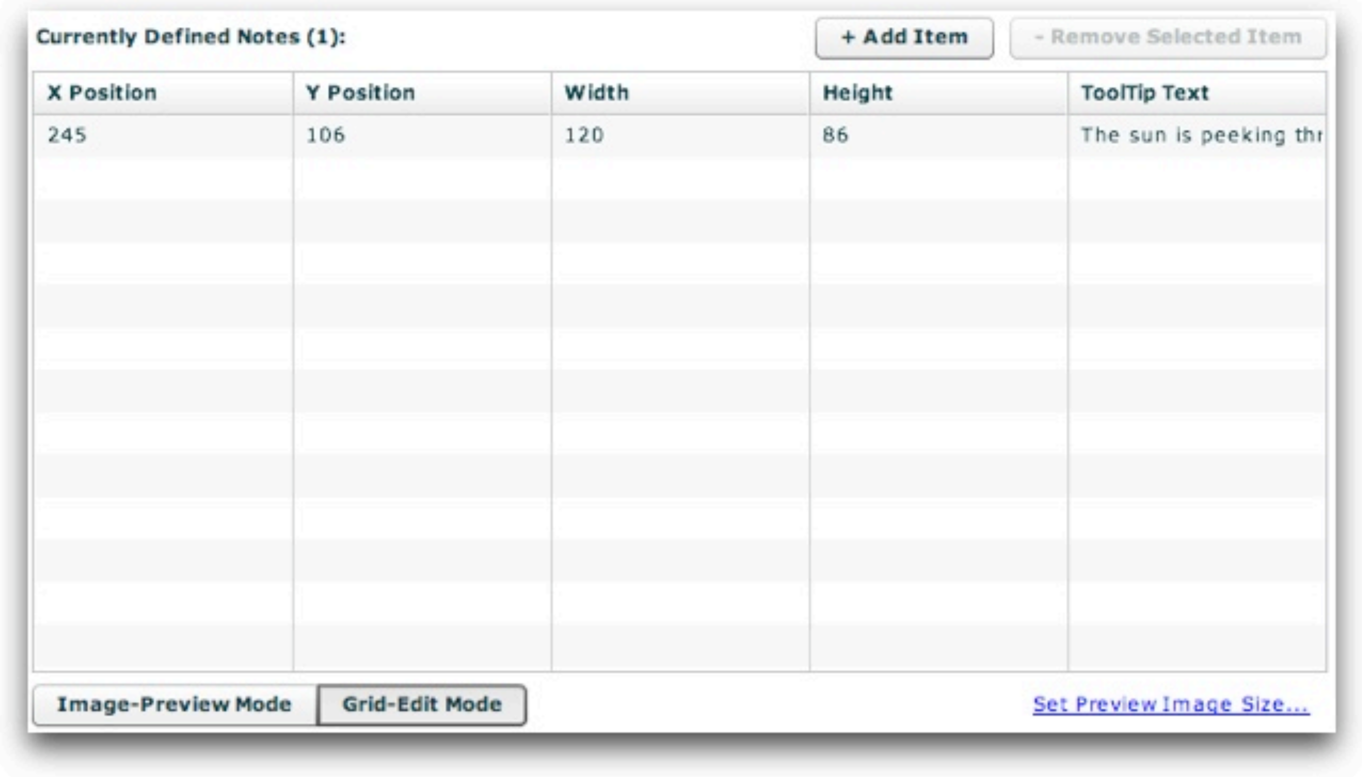


To resize a note, drag the bottom-right corner. A resize-handle will appear when you are over the appropriate area:



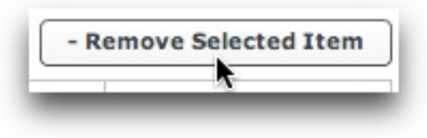
Using Grid-Edit Mode

While the Image-Preview Mode is good for adding notes, the Grid-Edit Mode is the place to remove notes or make other minor adjustments. After adding the note shown above, the Grid-Edit Mode section appears as follows:



X Position	Y Position	Width	Height	ToolTip Text
245	106	120	86	The sun is peeking thr

To edit any of the values in the grid, simply click on it; an edit box will appear. You can modify the X and Y positioning coordinates, the width or height, and the tooltip text which appears on hover. To add a new row, click the '+ Add Item' button. To remove a note, click on its row in the grid, then click the '-Remove Selected Item' button:



Advanced Customizations

If you are an advanced user, there are many overlay-note customizations which can be made by modifying XML settings. Detailed information regarding each of these settings can be found in the API reference for the below-mentioned classes.

To change how long the note boxes remain visible before automatically hiding, set the `noteHideDelay` property on the `IndividualSlide` style selector (see `com.dwuser.ss4.ui.IndividualSlide`). If you use a very large value for this setting, the boxes will remain visible after being activated instead of auto-hiding.

To customize the appearance of the boxes and their tooltips, you will use the `NoteBox` and `NoteBoxToolTip` style selectors respectively. For more information, see the `com.dwuser.ss4.ui.extras.NoteBox` and `com.dwuser.ss4.ui.extras.NoteBoxToolTip` classes.

To customize the behavior of the tooltips (e.g. the appearance delay and whether they track the mouse), you will use the `ToolTipAppearanceManager` style selector. For more information, see the `com.dwuser.ss4.managers.ToolTipAppearanceManager` class.

All overlay note instances are instances of the `com.dwuser.ss4.ui.extras.NoteBox` component. As such, the `<note ... />` nodes in the `<img>` nodes which define the overlay notes are actually `NoteBox` instances. They support most of the attributes available for the `NoteBox` component; see the class reference for available parameters.

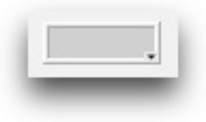
Special Editing Windows

This section covers special image-related editing windows which are only accessed through other areas of the wizard interface. You will generally want to view this section in context, when it is linked to from another topic.

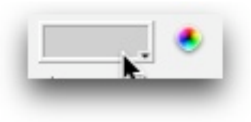
Using the External Wizard to Create and Edit Slideshows > Section-by-Section Usage Guide > Using the 'Images / Galleries' Section > Special Editing Windows > **The Color Picker + Color Mixer (Pro Version Only)**

The Color Picker + Color Mixer (Pro Version Only)

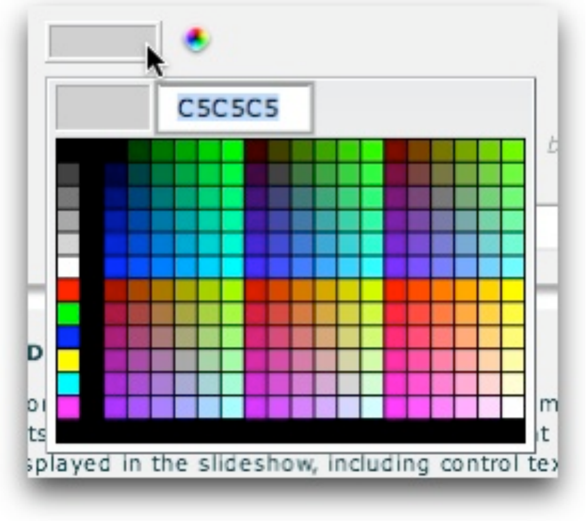
Throughout the wizard, there are various color-selection fields (color pickers) which appear like this:



When hovered over, you will see a color palette button as well:

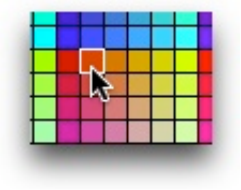


In both the basic version and the pro version, you can click on the visible swatch to open the swatch selection box:



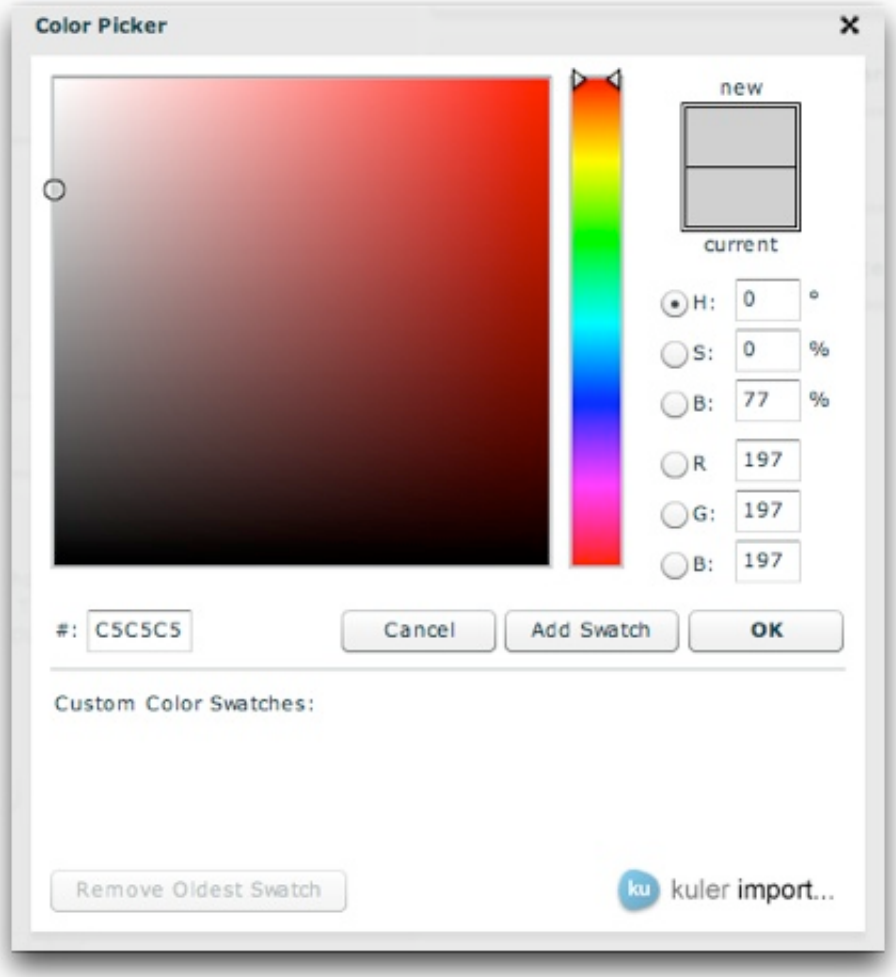
C5C5C5

You can either manually enter a hex color code in the box, or click on an existing color swatch to select it:



The Color Mixer Dialog

In the Pro version (learn about upgrading here), clicking on the small color palette button (to the right of color picker) will launch the Color Mixer dialog:



● H: 0 °

○ S: 0 %

○ B: 77 %

○ R 197

○ G: 197

○ B: 197

#: C5C5C5

Here, you can use various tools and sliders to select a color. You can drag the dropper around the large fade square. The radio buttons on the right allow you to pick or adjust a color based on one of the following aspects:

Property	Description
H	Allows color adjustments based on the hue
S	Allows color adjustments based on the saturation
B	Allows color adjustments based on the brightness

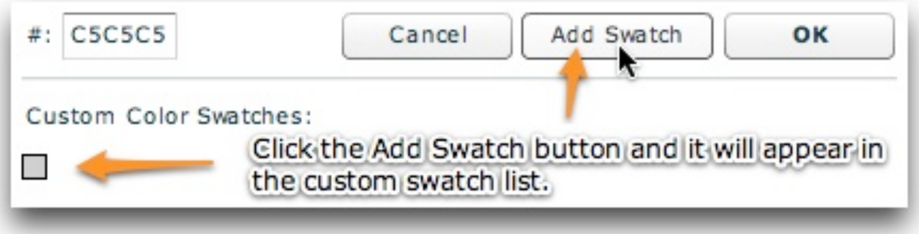
R	Allows color adjustments based on the Red channel
G	Allows color adjustments based on the Green channel
B	Allows color adjustments based on the Blue channel

It can sometimes take a little experimentation to determine the proper tool for your specific situation. You can also directly enter the hex value on the middle left.

When you're done making changes, click the **OK** button to exit the dialog. If you don't wish to apply your changes, click **Cancel** instead.

Using Swatches

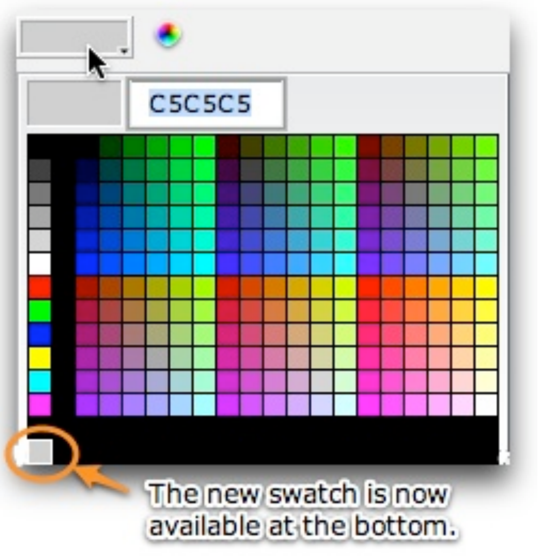
Swatches allow you to store commonly used colors for one-click access from the color swatch menu on the color picker. To add a color to the swatches list, click the **Add Swatch** button after selecting the color you want to store. It will appear in the Custom Color Swatches section:



#: C5C5C5

Click the Add Swatch button and it will appear in the custom swatch list.

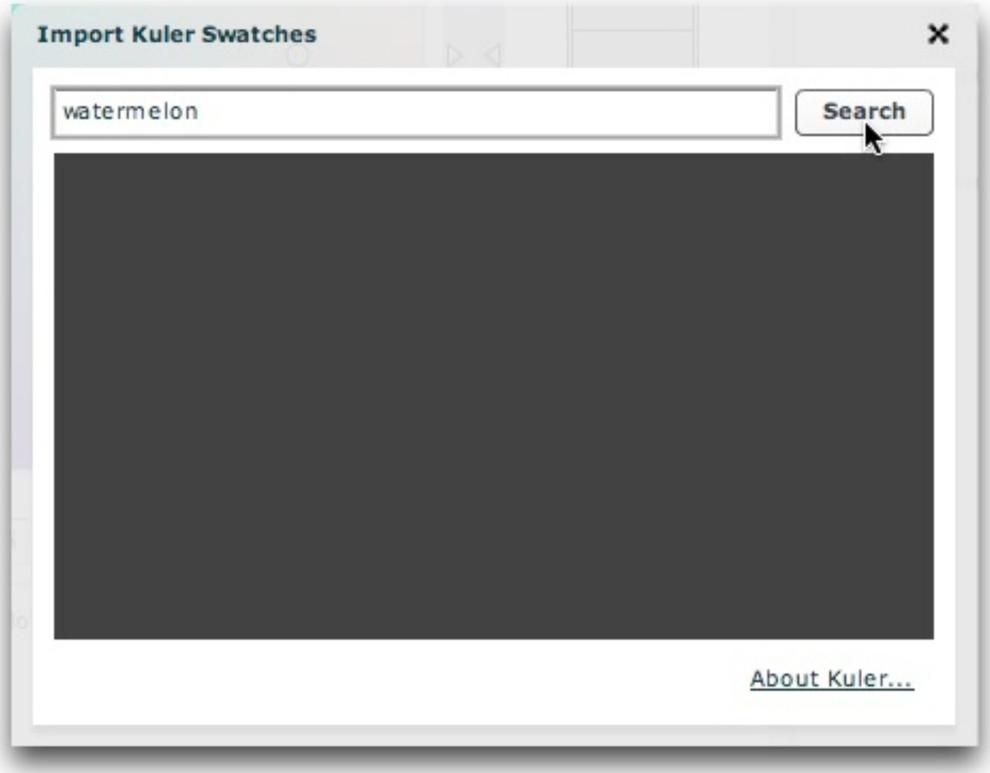
You can remove old swatches by using the **Remove Oldest Swatch** button. Once a swatch is in the custom swatch list, it can be accessed by clicking on it in the Color Mixer, or directly from the bottom of the color picker:



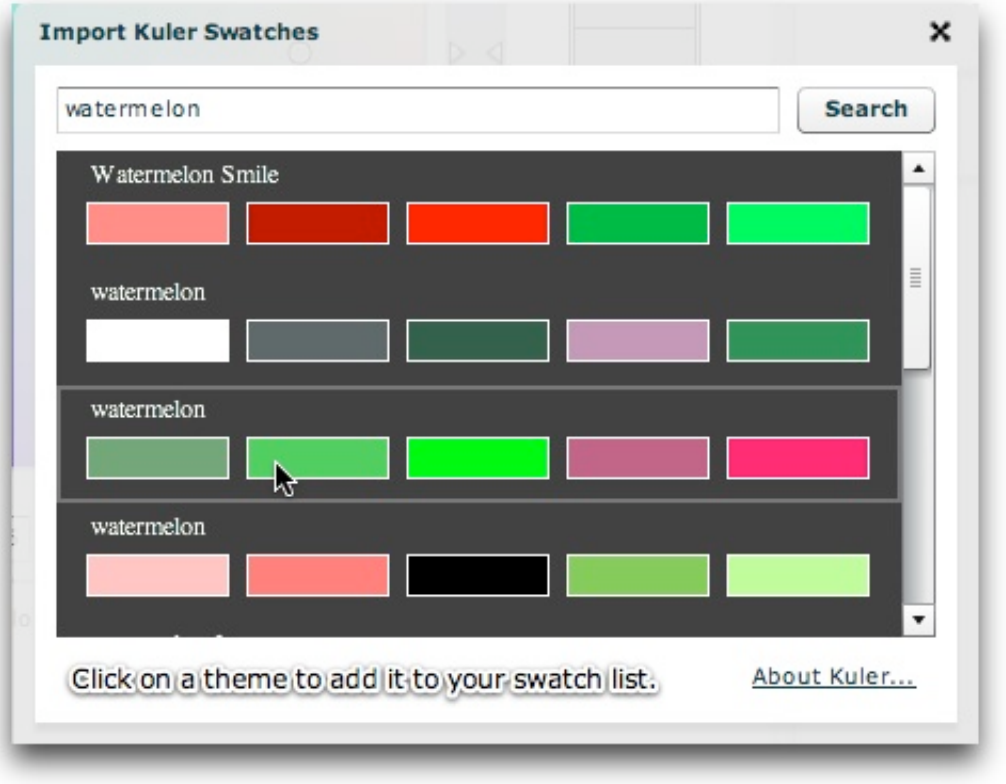
Importing Swatches from Kuler

Kuler, by Adobe, is a collaborative service which allows for free sharing of 5-color color schemes. You can use the Kuler Import feature to directly add these 5-color schemes to your swatch panel. To begin, click the Kuler Import button at the bottom of the Color Mixer dialog:

In the dialog, enter a search term and click Search:

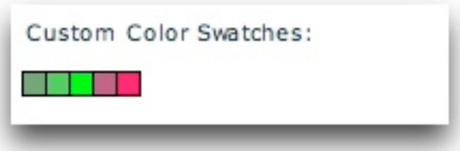


A list of matching color schemes will be loaded. To import a scheme, click on it:



Click on a theme to add it to your swatch list.

The five new colors will appear in your custom swatch list:



Click on any of the newly-added swatches to use that color.

The Autoplay Delay Editor

Autoplay delays are special strings which configure the delay between images / slides when autoplay is enabled. They can also enable automatic jumping to the image link URL associated with an image. Within the XML, these strings are specified through the `autoplayDelay` attribute. Because they can be a little complex to understand, there is a graphical interface to edit these delay strings. The first section below explains the string that the tool generates, then the second section shows how to use the tool for visual editing. Even if you don't plan on editing the delay strings manually, you probably still want to review the first portion so you understand the concepts.

Understanding the Structure of Autoplay Delay Strings

Note: This section is excerpted from the com.dwuser.ss4.managers.AutoplayManager documentation in the API reference. See that class for more detailed developer-targeted information about the autoplay feature.

To determine the delay between images, to disable autoplay on a specific image, or to automatically jump to a URL when an image is reached, you will modify the `autoplayDelay` attribute.

First, it's important to note the inheritance structure. When an image is opened, first the corresponding `<img ... />` node is checked for an autoplayDelay value. If one exists, it is used. Otherwise, the AutoplayManager next looks on the image's parent `<gallery>` node. If still none is found, it looks on the main `<galleries>` node. Thus, you can establish a default for all galleries or one specific gallery, while at the same time customizing the autoplayDelay on specific images as needed. Essentially, it just looks up the image's XML family tree for the first defined autoplayDelay attribute.

There are three different supported `autoplayDelay` formats:

- **`stop`** - Designed to be used on an `<img>` node; this `autoplayDelay` immediately pauses autoplay when the item is reached.
- **`jump`** - Designed to be used on an `<img>` node; this `autoplayDelay` causes the entire page to immediately reload and jump to the Image Link which has been defined for the image (in the XML, this is the `<img ... link="somewhere.htm" ... />` value). This is designed, for example, to allow the slideshow to automatically redirect once the last slide is reached.
- ***Comma-separated list, in order of priority*** - This is the format which will generally be used. See below for more information.

The comma-separated list format:

When you use the comma-separated list format, you'll construct a string with one or more of the following values in order of priority:

- *4000* (any numeric value in milliseconds) - Tells the slideshow to wait for *N* milliseconds before progressing to the next image.
- `kbComplete` - Tells the slideshow to wait for any KB effect which exists on the image to complete before progressing to the next image.
- `videoComplete` - Tells the slideshow to wait for any video which exists on the image to complete before progressing to the next image.
- `audioComplete` - Tells the slideshow to wait for any audio which has been added to the image to complete before progressing to the next image.

For example, if you wanted to always have a delay of 6 seconds between each slide, you'd use an autoplay delay of: `6000` . Suppose you had a narration attached to one of your images, and you wanted that slide to play the full narration before progressing. In this case, you'd use an `autoplayDelay` of: `audioComplete,6000` . This delay says essentially, "If there's audio on the slide, wait for it to complete before progressing. Otherwise, progress after six seconds."

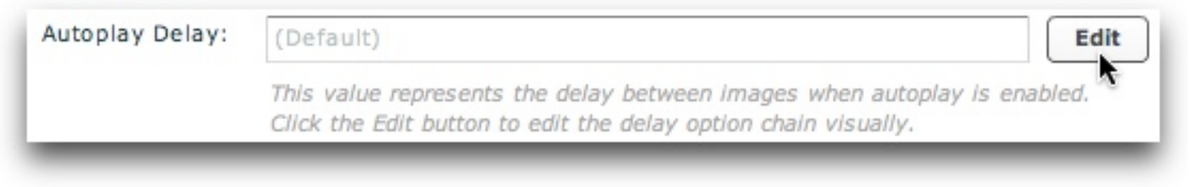
As a more complex example, suppose you had video on some slides, audio on other slides, and some KB effects alongside some of the audio. Because the audio is more important than the KB pan/zoom effect, it should always take priority. That is to say, the audio should be allowed to complete, even if the KB effect has already finished. So, you would want to use an `autoplayDelay` value of: `audioComplete,videoComplete,kbComplete,6000` . This delay says essentially, "If there's audio on the slide, wait for it to complete before progressing. If there's no audio but there is video, wait for the video to complete before progressing. If there's neither audio nor video but there is a KB effect, wait for the KB effect to complete before progressing. If there's no audio, video, nor KB effect, wait for six seconds before progressing."

Examples of specifying a global autoplayDelay, a gallery-specific autoplayDelay, and an image-specific autoplayDelay:

```
// Global
<galleries ... autoplayDelay="videoComplete,audioComplete,5000">

// Gallery-specific
<gallery ... autoplayDelay="kbComplete,6000">

// Image-specific - either when the KB effect ends, or 8 secs if no KB effect exists
<img ... autoplayDelay="kbComplete,8000">


// Image-specific - stops autoplay when the image is reached
<img ... autoplayDelay="stop">

// Image-specific - jumps to google when the image is reached
<img ... link="http://www.google.com/" autoplayDelay="jump">
```

Using the Visual Autoplay Delay Editor

If that previous section flew over your head, don't worry. There's a visual editor within the wizard to make this process easy. In various areas of the wizard (such as the Playback Settings main tab, the Gallery Properties dialog, or the Image Properties dialog), you'll see a box where the autoplay delay string is specified. Next to the box, there is an edit button which launches the visual delay editor:

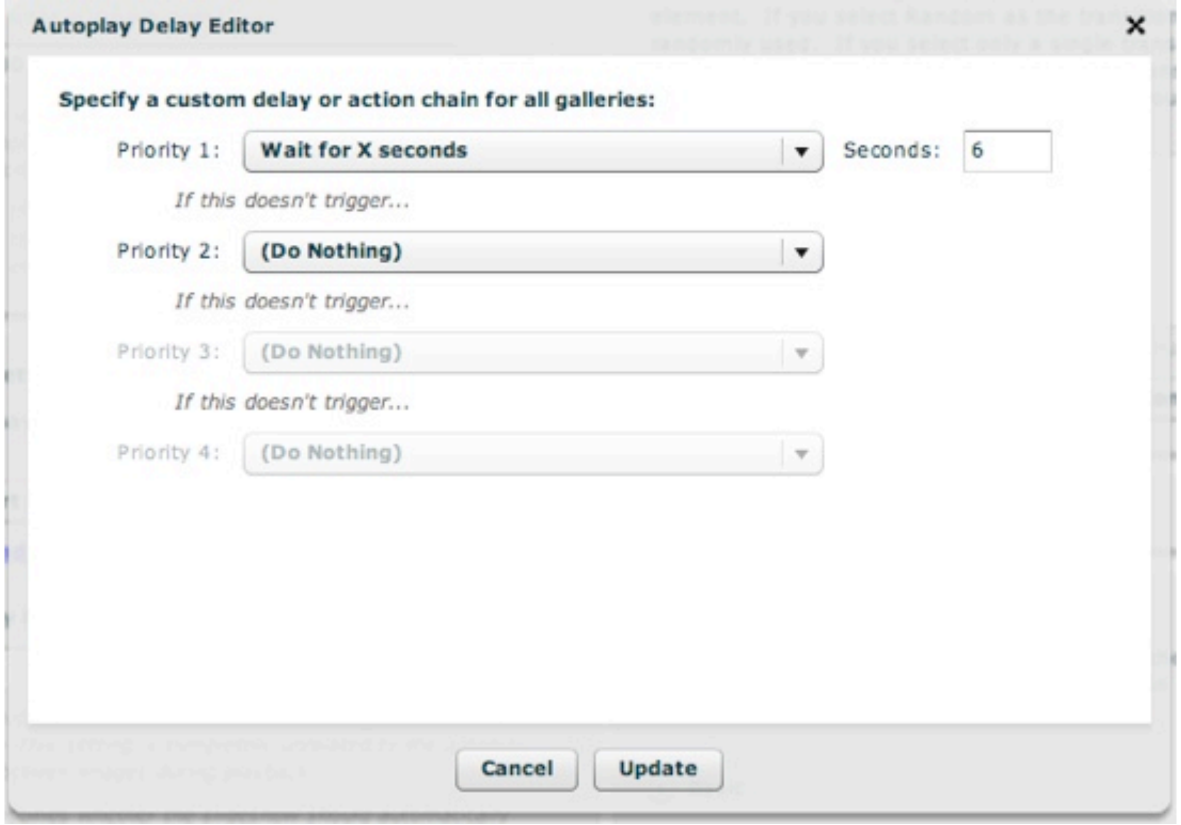


Autoplay Delay: (Default)

This value represents the delay between images when autoplay is enabled. Click the Edit button to edit the delay option chain visually.

There are minor differences between the global setting, the gallery-level setting, and the image-level setting. Image-related settings are only available on the image level, whereas the 'default to inherited value' setting is only available on the gallery and image level. The visual delay editor appears as follows for each level.

Global Setting (as accessed from the Global Configurationa and Playback Settings) tab:



Autoplay Delay Editor

Specify a custom delay or action chain for all galleries:

Priority 1: Wait for X seconds Seconds: 6

If this doesn't trigger...

Priority 2: (Do Nothing)

If this doesn't trigger...

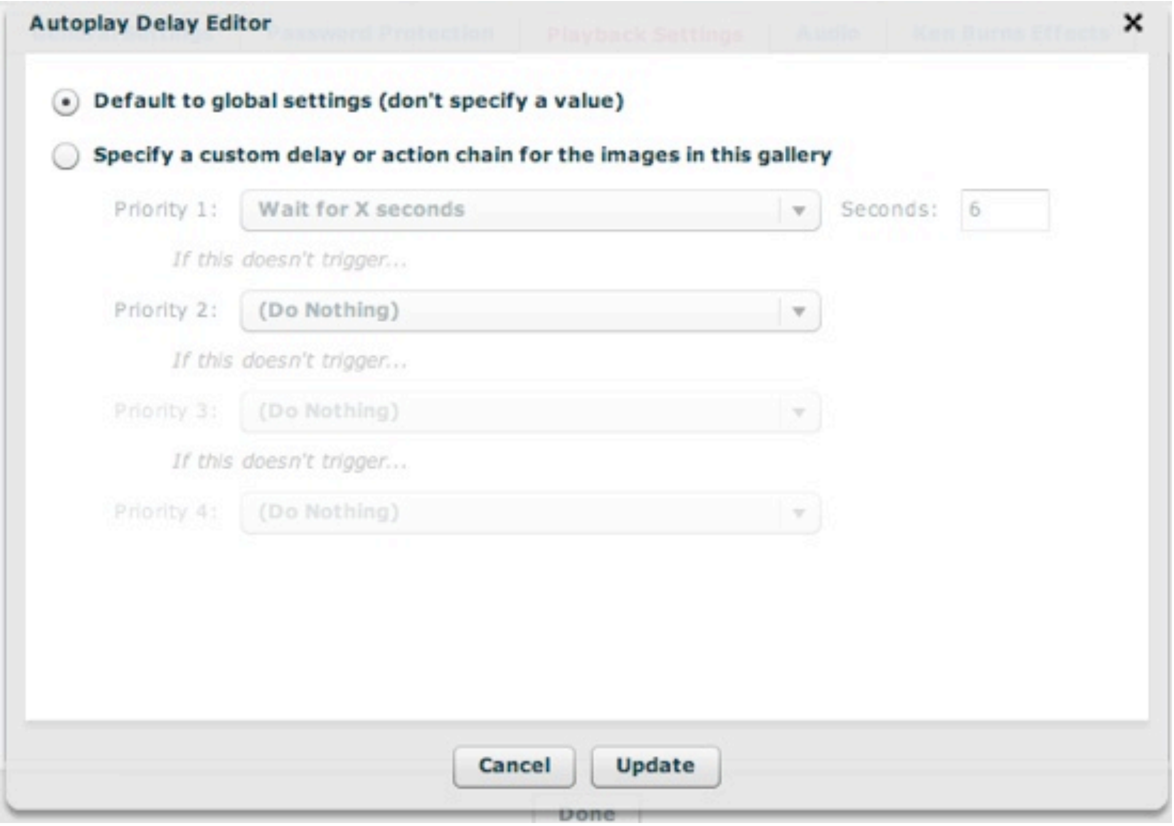
Priority 3: (Do Nothing)

If this doesn't trigger...

Priority 4: (Do Nothing)

For this level, you must specify a custom delay or action chain. See below for information on how to use the chain editor.

Gallery-Specific Setting (as accessed from the Gallery Properties dialog):



Autoplay Delay Editor

● Default to global settings (don't specify a value)

○ Specify a custom delay or action chain for the images in this gallery

Priority 1: Wait for X seconds Seconds: 6

If this doesn't trigger...

Priority 2: (Do Nothing)

If this doesn't trigger...

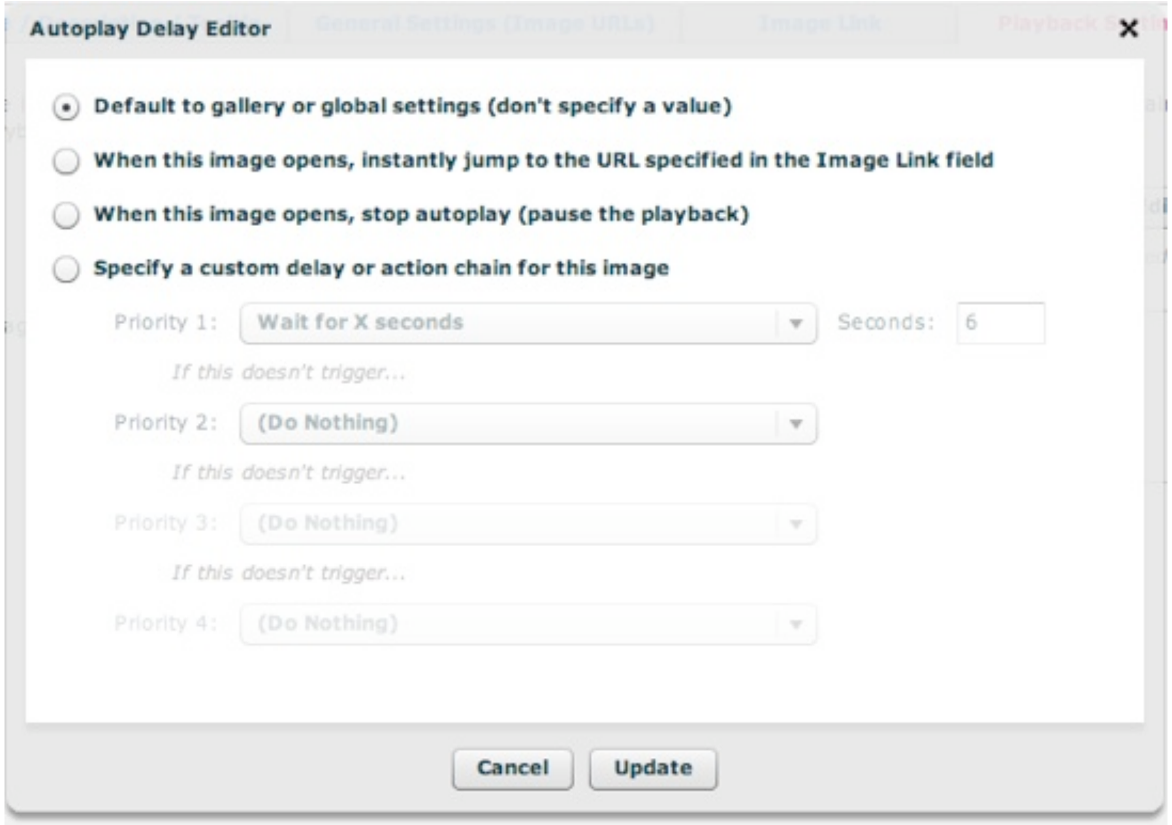
Priority 3: (Do Nothing)

If this doesn't trigger...

Priority 4: (Do Nothing)

For gallery-specific settings, you can default to the global settings (by using an empty value), or specify a custom delay or action chain (as explained below).

Image-Specific Setting (as accessed from the Image Properties dialog):



Autoplay Delay Editor

- ● Default to gallery or global settings (don't specify a value)
- ○ When this image opens, instantly jump to the URL specified in the Image Link field
- ○ When this image opens, stop autoplay (pause the playback)
- ○ Specify a custom delay or action chain for this image

Priority 1: Wait for X seconds Seconds: 6

If this doesn't trigger...

Priority 2: (Do Nothing)

If this doesn't trigger...

Priority 3: (Do Nothing)

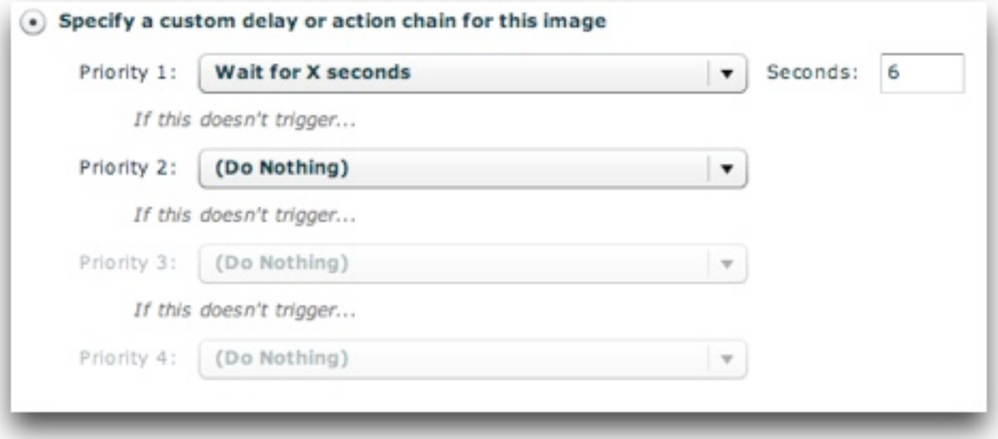
If this doesn't trigger...

Priority 4: (Do Nothing)

For image-specific settings, you have several special options. You can default to the global or gallery settings (by using no value). You can also tell the slideshow to jump to the URL specified in the Image Link field as soon as the image is reached. Or, you can tell the slideshow to pause playback as soon as the image is reached. Finally, you can also specify a custom delay or action chain (as explained below).

Constructing a Custom Delay or Action Chain

This is the area you will use to specify a specific timed delay or to tell slideshow to wait for a specific event before continuing to the next image:



● Specify a custom delay or action chain for this image

Priority 1: Wait for X seconds Seconds: 6

If this doesn't trigger...

Priority 2: (Do Nothing)

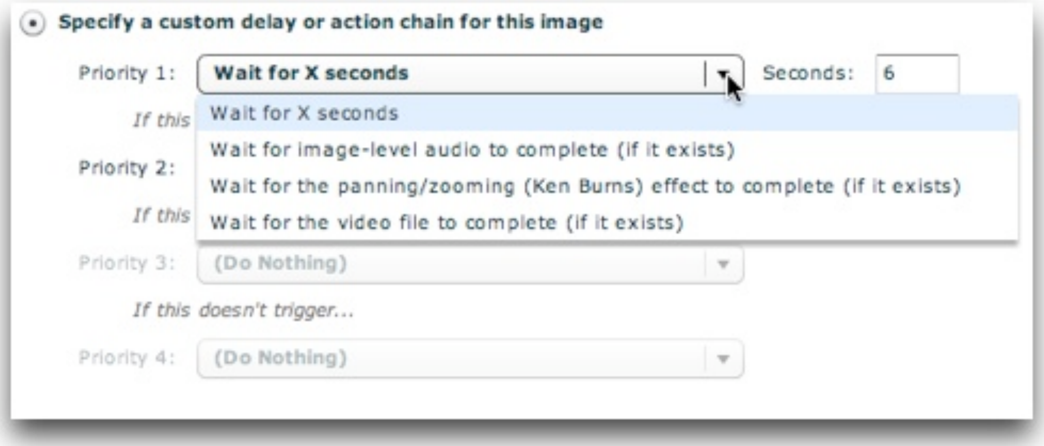
If this doesn't trigger...

Priority 3: (Do Nothing)

If this doesn't trigger...

Priority 4: (Do Nothing)

You construct a "chain" of priorites, with the highest priorites coming first. You can choose between a timer completing ('Wait for X seconds'), waiting for any image-level audio to complete, waiting for any video to complete, or waiting for any panning/zooming (Ken Burns) effect to complete. If any of these items are not applicable for a specific image, that item is skipped and the next item in the chain is checked for applicability until an item can be applied. If none is found, autoplay won't progress. The Priority 1 option looks like this:



● Specify a custom delay or action chain for this image

Priority 1: Wait for X seconds Seconds: 6

Wait for X seconds
Wait for image-level audio to complete (if it exists)
Wait for the panning/zooming (Ken Burns) effect to complete (if it exists)
Wait for the video file to complete (if it exists)

If this

Priority 2:

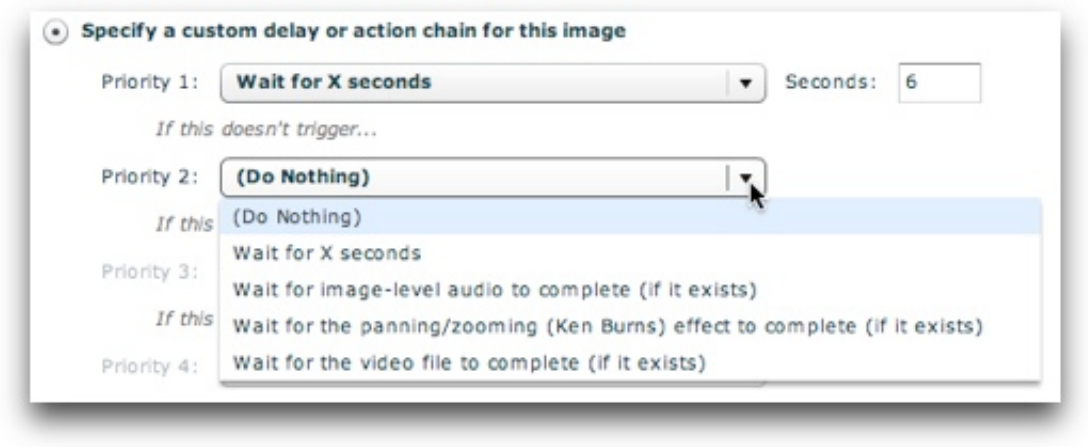
If this

Priority 3: (Do Nothing)

If this doesn't trigger...

Priority 4: (Do Nothing)

For subsequent priority levels (Priority 2, 3, and 4), you can either choose Do Nothing or you can provide another fallback option.



● Specify a custom delay or action chain for this image

Priority 1: Wait for X seconds Seconds: 6

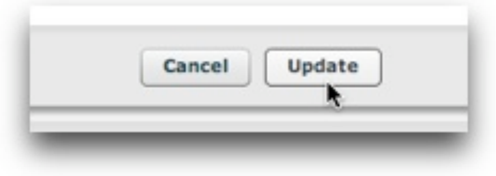
If this doesn't trigger...

Priority 2: (Do Nothing)

If this
Priority 3:
If this
Priority 4:
(Do Nothing)
Wait for X seconds
Wait for image-level audio to complete (if it exists)
Wait for the panning/zooming (Ken Burns) effect to complete (if it exists)
Wait for the video file to complete (if it exists)

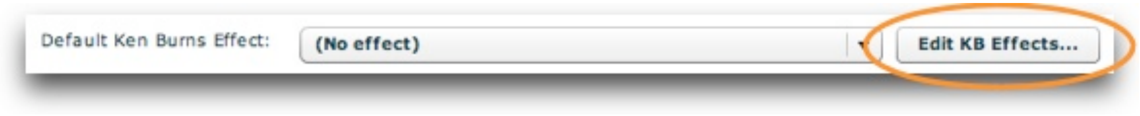
If on any priority level you select 'Wait for X seconds,' you'll need to enter a value in the Seconds box to the right of that level. Note - while you can choose the same behavior for multiple priority levels, it has no effect.

Once you've finished making changes, click the Update button at the bottom:



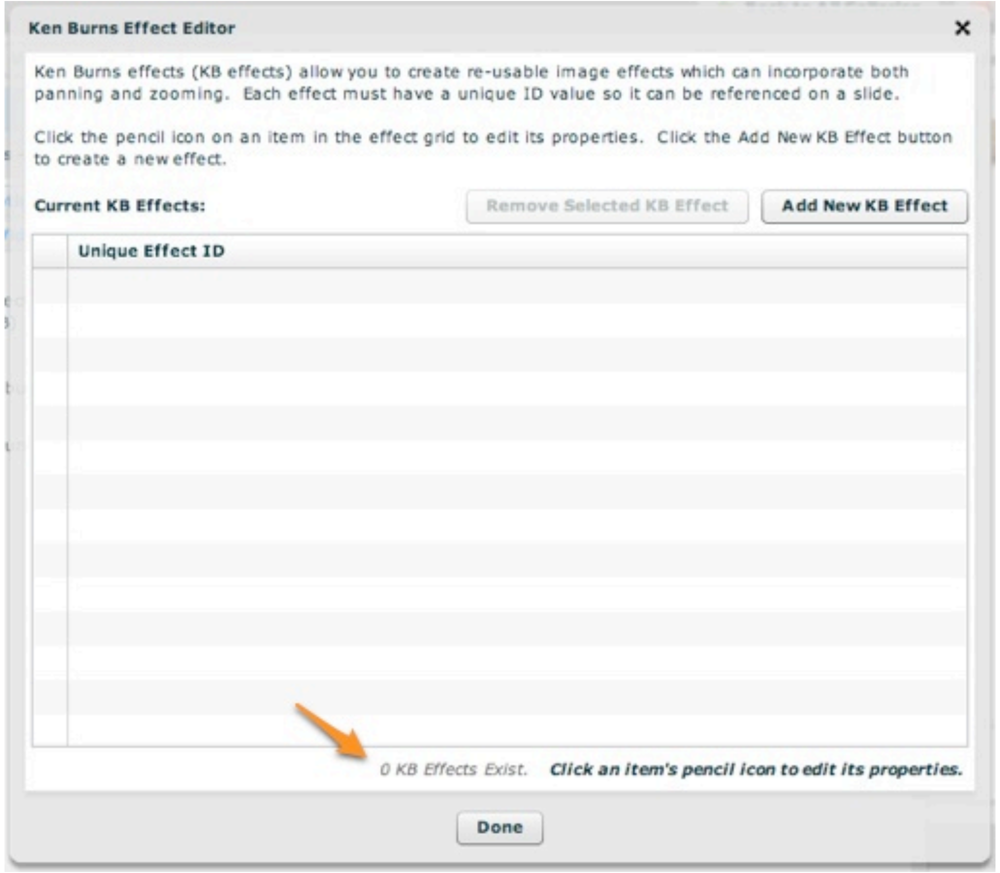
The Ken Burns Effect (Panning / Zooming) Editor

Ken Burns effects (KB effects) allow you to create re-usable image effects which incorporate both panning and/or zooming. Because you will often use the same effect on multiple images, KB effects are defined as standalone effects, each with a unique ID value. Then, when you want to use a specific effect for an image or gallery, you will reference the effect's ID value. Because all KB effects are stored centrally, you will use the KB Effect Editor dialog to manage the effects. The editor is accessed through either the Gallery Properties dialog (which establishes a default for the entire gallery) or the Image Properties dialog (which establishes an image-specific effect):



Default Ken Burns Effect: (No effect)

The editing dialog appears initially as follows:



Ken Burns Effect Editor

Ken Burns effects (KB effects) allow you to create re-usable image effects which can incorporate both panning and zooming. Each effect must have a unique ID value so it can be referenced on a slide.

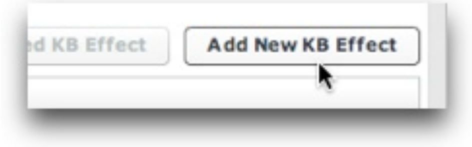
Click the pencil icon on an item in the effect grid to edit its properties. Click the Add New KB Effect button to create a new effect.

Current KB Effects:

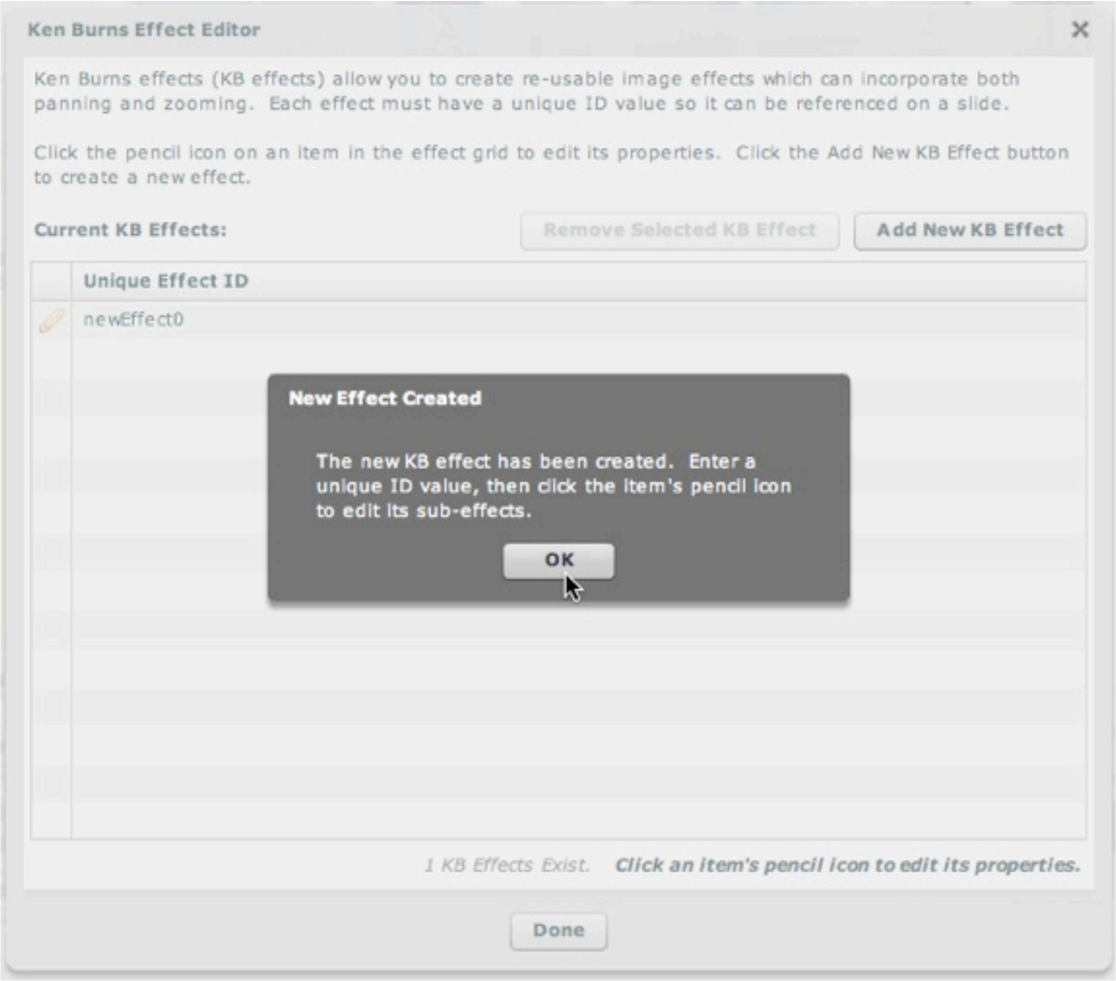
Unique Effect ID

0 KB Effects Exist. *Click an item's pencil icon to edit its properties.*

So far, no effects have been created. To get started, click the 'Add New KB Effect' button above the grid:



The new empty effect will be created and you will be notified:

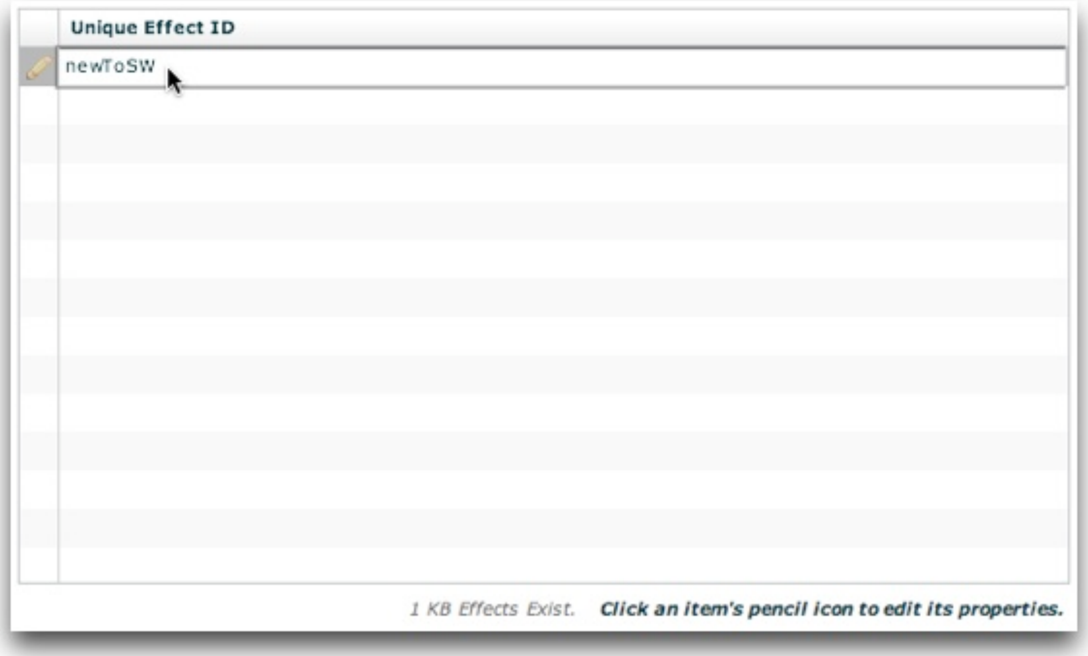


Ken Burns effects (KB effects) allow you to create re-usable image effects which can incorporate both panning and zooming. Each effect must have a unique ID value so it can be referenced on a slide.

Click the pencil icon on an item in the effect grid to edit its properties. Click the Add New KB Effect button to create a new effect.

The new KB effect has been created. Enter a unique ID value, then click the item's pencil icon to edit its sub-effects.

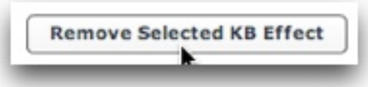
As noted in the alert box, you must now give the effect a unique ID value with which it can be referenced. While the default value (`newEffect0` in this case) can be retained, it is better to use a name which has significance, e.g. `NWtoSE` for an effect which goes from top-left to bottom-right. To edit the ID value, click on its ID in the effect grid:



Unique Effect ID
newToSW

1 KB Effects Exist. *Click an item's pencil icon to edit its properties.*

If you want to remove an effect, click on it to select it, then click the 'Remove Selected KB Effect' button:



Setting Effect Properties

Now that a new effect has been created, you're ready to configure the effect's properties. Each effect has two points which you configure: its starting point (the initial position), and its ending point (the final position). Each of these points has three properties:

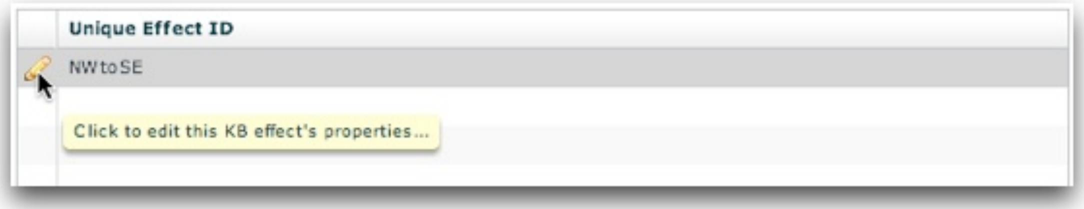
- **X-coordinate** - This value represents the horizontal position relative to the center of the image. For example, -100 would represent a point 100 pixels to the left of the image's center. In this case, the left side of the image would be visible at this point.
- **Y-coordinate** - This value represents the vertical position relative to the center of the image. For example, 100 would represent a point 100 pixels below the image's center. In this case, the bottom edge of the image would be visible at this point.
- **Zoom factor** - This value represents the scaling factor applied to the image, with 1 being normal size (100%). For example, 1.2 represents 120%.

If either the relative X or Y coordinate value is outside the image's actual dimensions (e.g. an X value of -300 on an image which is only 400 pixels wide, where -200 is the smallest possible value), the value will automatically be adjusted to the maximum possible value (-200 in this example). Because of this, you can use a very large X or Y value to ensure that an effect always starts at the edge of an image (e.g. 1000 or -1000).

Unless you are using images larger than the visible area, you will not want to use a zoom factor of less than one (which represents zooming out).

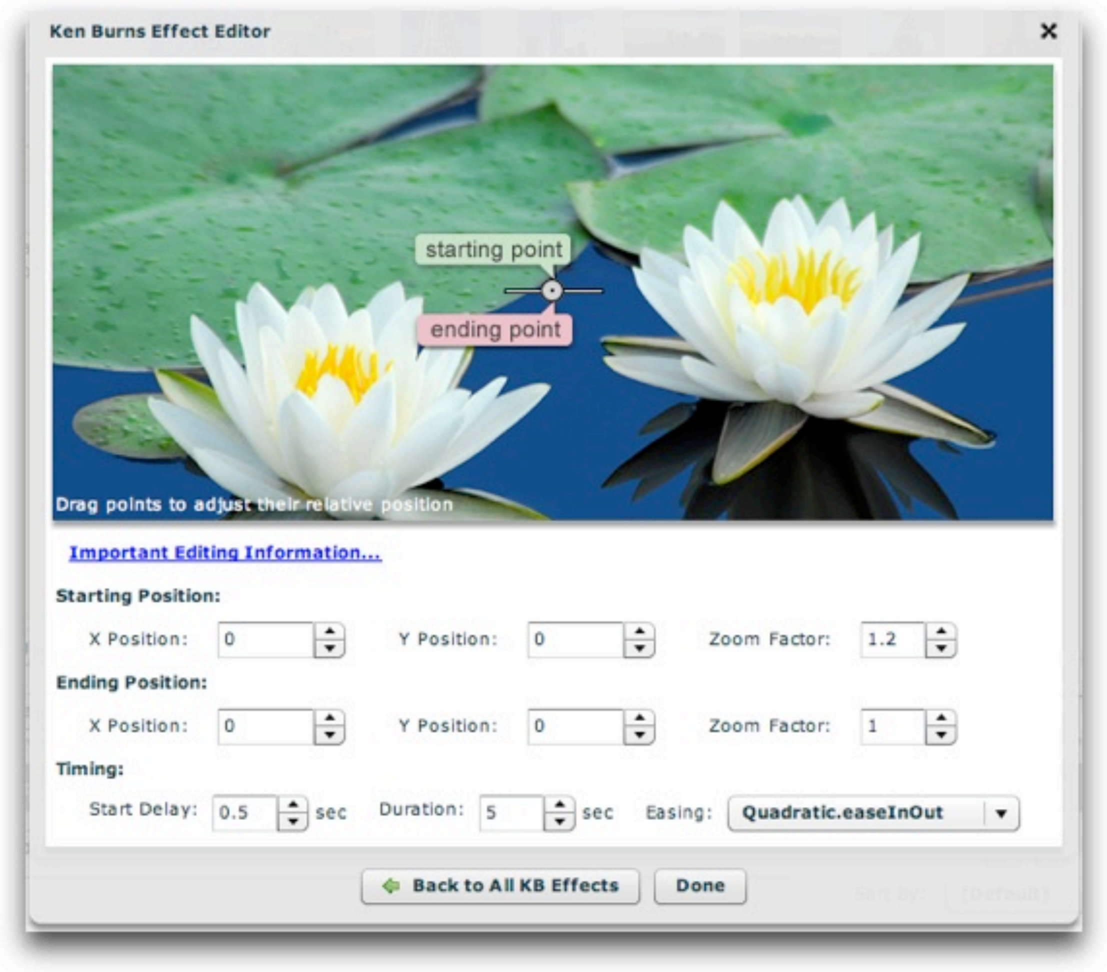
When an image opens, it will be initially displayed at the starting point. If any **start delay** has been applied, it will stay there until the delay timer completes. Then, over a period specified by the **duration**, it will pan and/or zoom to the ending position. The **Easing** parameter controls the easing effect (e.g. easing in, out, or both) applied to the transition. If you specify the same value for any property at both the starting and ending points, no transition will take place in that dimension. For example, if you specify a starting and ending zoom factor of 1 no scaling/zooming will occur.

To edit an effect's properties, click the pencil icon to the left of the effect's ID value:



Unique Effect ID
NWtoSE

This will transition you to the effect property editor view:


Ken Burns Effect Editor
×
starting point
ending point
Drag points to adjust their relative position
Important Editing Information...
Back to All KB Effects
Done

Starting Position:

X Position: 0 Y Position: 0 Zoom Factor: 1.2

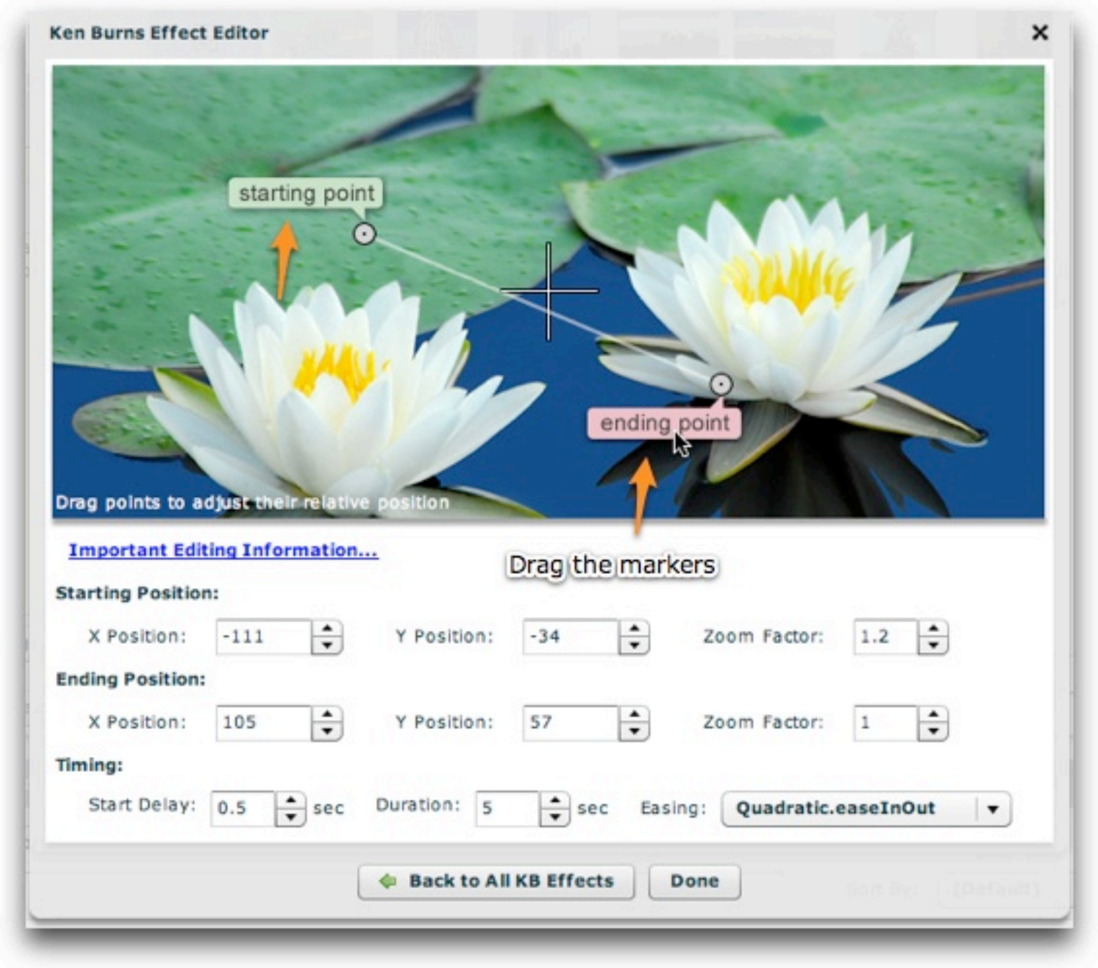
Ending Position:

X Position: 0 Y Position: 0 Zoom Factor: 1

Timing:

Start Delay: 0.5 sec Duration: 5 sec Easing: Quadratic.easeInOut

The image represents an example slide. The crosshairs in the center represent the exact center. You can manually edit the starting, ending and timing properties at the bottom, or you can visually edit at the top by dragging and dropping the starting and ending markers. For this top-left to bottom-right example effect, you might drag the markers into a position such as this:



Starting Position:

X Position: -111 Y Position: -34 Zoom Factor: 1.2

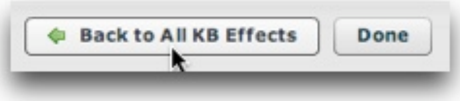
Ending Position:

X Position: 105 Y Position: 57 Zoom Factor: 1

Timing:

Start Delay: 0.5 sec Duration: 5 sec Easing: Quadratic.easeInOut

When you're finished editing an effect's properties, click the 'Back to All KB Effects' button to return the main effect grid (to, for example, create more effects); or, click Done to exit the KB Effect Editor dialog:



Using the 'External Image Data' Section

This section of the manual covers the integration of dynamic data into slideshows. To get a full picture of how to use dynamic data, you should view all of the subtopics in this section.

Understanding How Dynamic Data Works

When you create a slideshow in the wizard, you manually create galleries and add images to these galleries. For most instances, this is fully sufficient. However, sometimes as an advanced user or developer you may want to be able to populate galleries -- or even define galleries -- on the fly, based on changing ("dynamic") information. This can include images in a given directory, database contents (via a recordset), or data from other external sources such as Flickr. Integration of this type of data is accomplished through the use of Dynamic Data Sources. These are special URL's which are passed to the slideshow. The slideshow loads the data from these URL's, transforms it to the slideshow image-data format if necessary, and adds it to the static XML configuration file as if it had been added in the wizard manually.

One of the most common uses of dynamic data is to allow a client to manage their slideshow images directly from a third-party service such as Flickr. The client uploads images to Flickr and specifies title and description/caption information. As soon as the client makes changes, these are reflected in the slideshow live on their site. Because you can integrate specific Flickr photoset, the client can manage multiple slideshows from a single account. More information on this is available in the 'Managing Dynamic Data in the Wizard' topic, which details Flickr API integration.

If you're comfortable with server-side programming, you can also write your own scripts which dynamically generate image data. For more information on this and example code, see the 'Creating Data Sources + Example Code' topic.

Loading from Different Domains

If you load dynamic data from an external source (a different domain), you may be affected by Flash Player cross-domain loading security restrictions. To learn more about this, see the Understanding Flash Player Cross-domain Loading Restrictions topic. Note that some sources, such as Flickr, have unrestricted policy files, so you won't be affected by these restrictions.

Passing Parameters to Dynamic Data Sources

Especially if you are writing and using your own dynamic data source scripts, you may need to pass parameters to the dynamic data source file. For example, you might need to display images of a certain property based on the property being displayed on the slideshow page. The slideshow might be in `property_details.php?property_id=12345` . In this case, you would need to let the dynamic data script know that it should retrieve and output image information for property `12345`. To do this, you must configure your dynamic data script to use a passed `property_id` parameter, then mirror this value to the dynamic data source URL. In your slideshow HTML code, you would need to locate the `xml:'...'` value, which tells the slideshow where to load all data from. This value will contain the URL of the dynamic data source you added in the wizard. At the end of this URL, add the appropriate code to output the `property_id` parameter. For example:

```
... xml:'v4flashslideshow/slideshow_dataN.xml,slideshow_data.php?property_id=<?php
echo($_GET["property_id"]); ?>', ...
```

Note that this example has no data value validation, and should NOT be used in production.

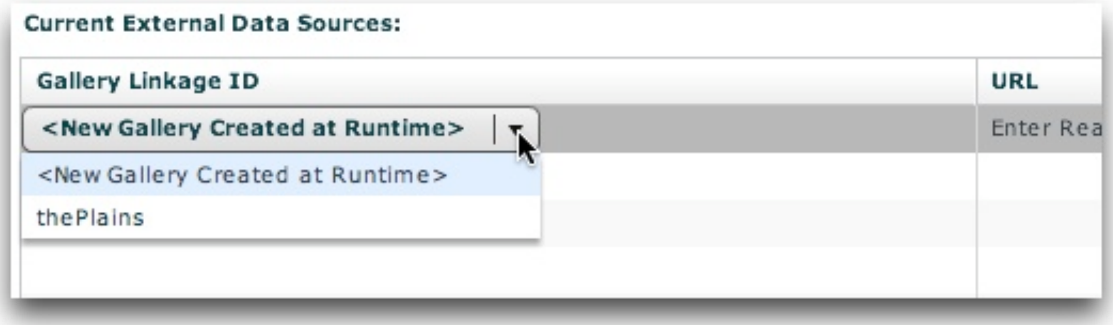
The point is that you must mirror any needed parameters to the data source through its URL. These parameters will then be accessible from the dynamic data source file.

Supported Dynamic Data Formats

There are several formats which the slideshow can load and either import directly or transform and then import. This topic details each of these formats.

Importing Dynamic Data: New Gallery or Existing Gallery?

When using any format except the multiple-gallery format, there are two options for the way the data is imported. It can either be added to a new gallery which is created at runtime, or it can be added to an existing gallery which you defined in the wizard. The latter method allows fine-grained control over gallery properties, such as the gallery title, caption, and thumbnail image, as well as default effects. To associate a dynamic data source with an existing gallery, you must specify an ID value for that gallery in the Gallery Properties dialog. Then, in the External Image Data section of the wizard, select that gallery as the data source's associated gallery:



Current External Data Sources:

Gallery Linkage ID	URL
<New Gallery Created at Runtime>	Enter Rea

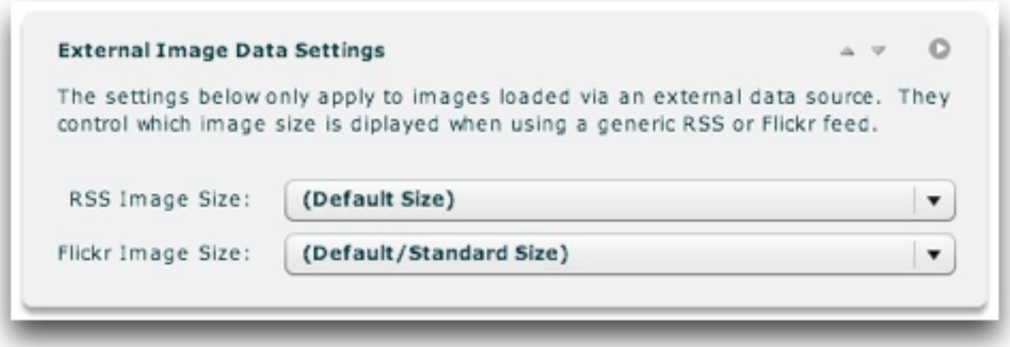
If editing the dynamic data source URL directly in the XML file, a special value is prepended to the start of the URL to indicate that it should be added to an existing gallery. This value has following format within the slideshow HTML code:

```
... xml:'v4flashslideshow/slideshow_dataN.xml,{gallery:linkageID}dynamic_data_source.php', ...
```

In other words, a value of `{gallery:linkageID}` is prepended to the URL, where *linkageID* is the linkage ID you specified for the gallery in the wizard as mentioned above.

Default RSS Feed and Flickr Image Size

When using a Media RSS feed or a Flickr feed or Flickr API data, there may be multiple image sizes available. You can choose the default size in the main 'Global Configuration and Playback Settings' tab. Expand the 'External Image Data Settings' pod, and change the default sizes as desired:



External Image Data Settings

The settings below only apply to images loaded via an external data source. They control which image size is diplayed when using a generic RSS or Flickr feed.

RSS Image Size: (Default Size)

Flickr Image Size: (Default/Standard Size)

For more information, see the Using the 'Global Configuration and Playback Settings' Section portion of the manual.

Single Gallery Format

The most common format used allows the inclusion of a single externally-defined gallery. This format is used for data sources which you create. To use this format, your dynamic XML document will have the same format as an individual gallery within the XML configuration file. (For detailed information, see *The <galleries> section - Image Data* within the Developer Reference.) Here is an example:

```
<gallery title="Gallery Title" description="Gallery description goes here"
thumb="path/to/thumbnail_image.jpg">
    <img src="path/to/image1.jpg" thumb="path/to/thumb_image1.jpg" title="Title for the first image"
description="The description..." />
    <img src="path/to/image2.jpg" thumb="path/to/thumb_image2.jpg" title="Title for the second image"
description="The description..." />
    <img src="path/to/image3.jpg" thumb="path/to/thumb_image3.jpg" title="Title for the third image"
description="The description..." />
</gallery>
```

In this example, all of the gallery information is optional if you link the data source with an existing gallery. Only the *src* attribute is required on the <img> nodes; see the developer reference for more information.

Note: There is a template for this format available from within Dreamweaver by going to File > New... > Page from Sample > XML Flash Slideshow v4. For more information, see the Creating Data Sources + Example Code topic in this section.

Multiple Gallery Format

A format similar to the Single Gallery Format is the Multiple Gallery Format. It includes one or more `<gallery>` nodes in the exact same format as shown above, but it wraps them in a `<galleries>` node. This allows you to specify multiple dynamic galleries at runtime. Because multiple `<gallery>` nodes are used, associating this format with an existing gallery has no effect; you must pass all gallery-specific information directly in the dynamic XML. In the end, the data source file will look like the following example, with each truncated <gallery> node being in the format as shown above for the Single Gallery Format:

```
<galleries>

    <gallery title="First Gallery Title" description="Gallery description goes here"
thumb="path/to/thumbnail_image1.jpg">
        ...first gallery images here...
    </gallery>

    <gallery title="Second Gallery Title" description="Gallery description goes here"
thumb="path/to/thumbnail_image2.jpg">
        ...second gallery images here...
    </gallery>


    <gallery title="Third Gallery Title" description="Gallery description goes here"
thumb="path/to/thumbnail_image3.jpg">
        ...third gallery images here...
    </gallery>


</galleries>
```

Media RSS Format

Data can be loaded from RSS feeds which use the standard Media RSS feed format (such as Flickr Feeds or SmugMug or Picasa). When adding feeds as dynamic data sources, make sure that the URL starts with `http://` , not `feed://` (which is a pseudo-protocol used to tell browsers that the URL is a feed). If using a Flickr feed, make sure that the URL includes `format=rss_200` - this forces the feed to be in Media RSS 2.0 format. Note that while feed data can be loaded from some services (such as SmugMug), if these services do not allow cross-domain loading the slideshow won't be able to load or display photos. For more information, see the notes on cross-domain loading in this topic; you may want to check with the provider to find out if they allow cross-domain loading.

For more information on using Flickr RSS Feeds, see the Flickr Integration topic.

Flickr API Format

Data can be loaded directly from the Flickr API in Flickr's REST data format (the default format). The API call must be one which returns a list of photos inside a `<photos>` node. For more information, see the Flickr Integration topic. Because the Flickr API only returns the title of photos in the photo lists, you can pass additional data in the format using the built-in transformation support. For more information, see the XML Transformation / Passing Custom Parameters topic.

ATOM Format (Unsupported!)

There was originally limited ATOM support in early builds. However, this feature was dropped later on due to better formats being commonly available. ATOM feeds can be experimentally used, but there is no support for this format.

Using the External Wizard to Create and Edit Slideshows > Section-by-Section Usage Guide > Using the 'External Image Data' Section > **XML Tranformation / Passing Custom Parameters**

XML Transformation / Passing Custom Parameters

When using dynamic data sources, you often won't have control over all of the fields which are passed. Or, only some of the fields you want to use will be available. In these cases, you will need to use the built-in XML transformation capabilities to convert the XML into the format you need.

The transformation capabilities in this section show how you can pass a description in the as well as other image parameters, such as a video file or image tooltip. These techniques apply also to RSS feeds, where you can specify both a title and description but no other image property fields.

Passing a Description in the Title

This technique is used with external data sources where you are only able to pass a single parameter -- the title. You can embed a description in the title by using the following format:

> The Image Title**{desc}**The image description is here.

Wherever {desc} is placed in the title, the string will be split, with the first half used as the title and the latter half used as the description/caption.

Note: {capt} has the same effect as {desc}, but {desc} is the preferred value.

The custom-parameter technique shown below (for passing special parameters in the description) can be used in conjunction with this method, allowing for the title, description, and other parameters to all be passed in the title.

Passing Custom Parameters in the Description / Caption

In instances where you only can pass image title and description information (such as RSS feeds or Flickr API data), you may still want to pass XFS-specific image parameters, such as a video file or tooltip. This can be done by including special values of the following format within the description/caption value:

> {setAtt:***attName***=***attValue***}

Replace `attName` with the name of the attribute you want to set, and `attValue` with the new value of the attribute. For example, if you wanted to set the image's tooltip (via the `toolTip` attribute) to be 'A cool tooltip!', you would use the following value within the description in the dynamic data source:

```
{setAtt:toolTip=A cool tooltip!}
```

This can be used with any attribute. Even though it's not recommended, you could even modify the `thumbnail` or `src` attributes (which specify the image URL's) if you don't like the default values which are coming through. Here's an example of the above text as part of an actual description:

```
This image has a very unique description.  This is the unique description.{setAtt:toolTip=A cool tooltip!}
```

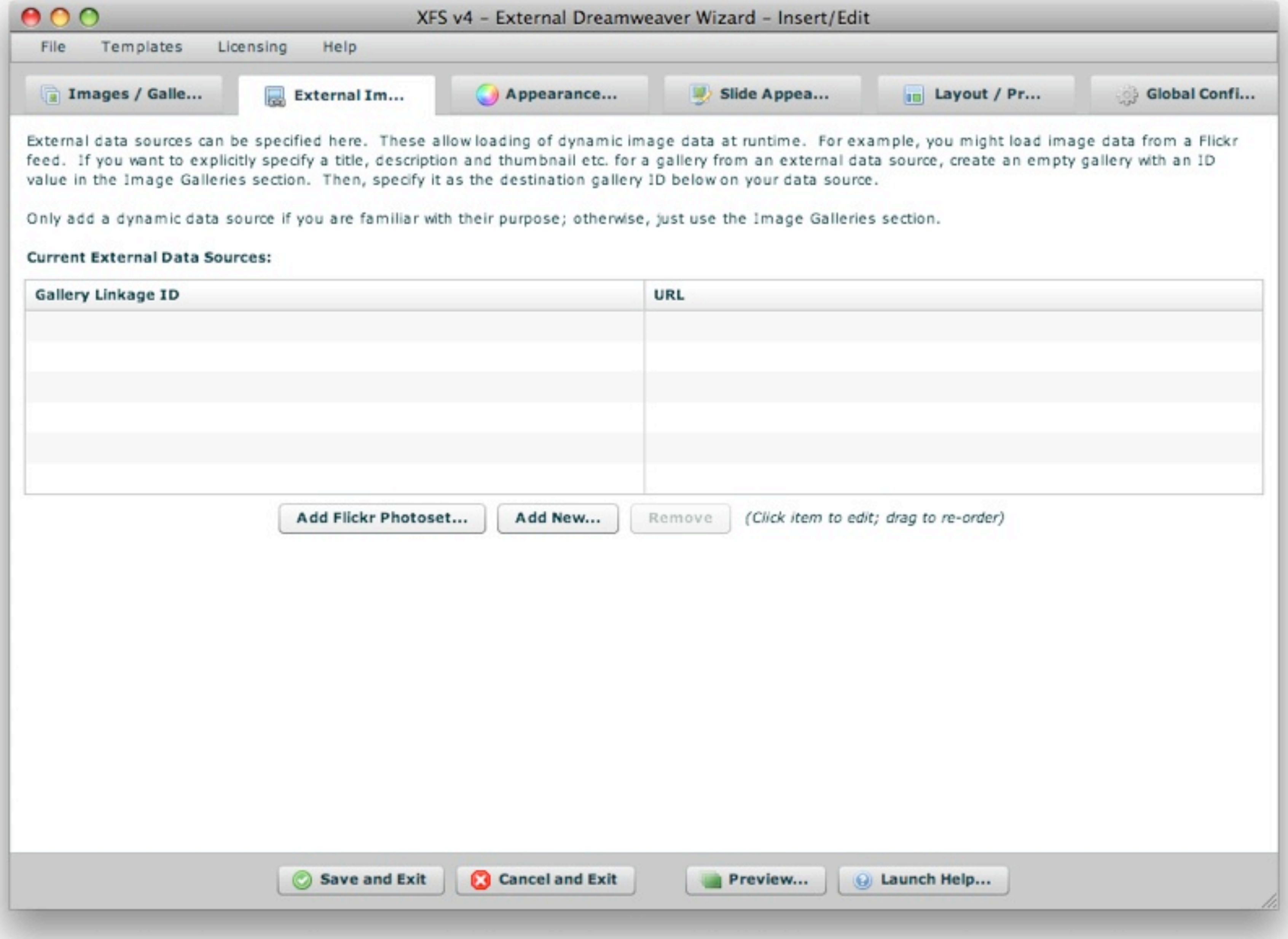
Tip: To learn about all of the supported XML attribute names, browse a generated XML configuration file or see the Developer Reference section of the manual.

Using the External Wizard to Create and Edit Slideshows > Section-by-Section Usage Guide > Using the 'External Image Data' Section > **Managing Dynamic Data in the Wizard**

Managing Dynamic Data in the Wizard

Note: Make sure you read this topic as a part of reading the entire Using the 'External Image Data' Section portion of the manual. Reading this individual topic in isolation doesn't give the full overview of how dynamic data works.

Once you understand the concept of dynamic data sources and have either created or located a data source, you're ready to add it in the wizard. This is done in the main 'External Image Data' tab of the wizard. It appears as follows:



External data sources can be specified here. These allow loading of dynamic image data at runtime. For example, you might load image data from a Flickr feed. If you want to explicitly specify a title, description and thumbnail etc. for a gallery from an external data source, create an empty gallery with an ID value in the Image Galleries section. Then, specify it as the destination gallery ID below on your data source.

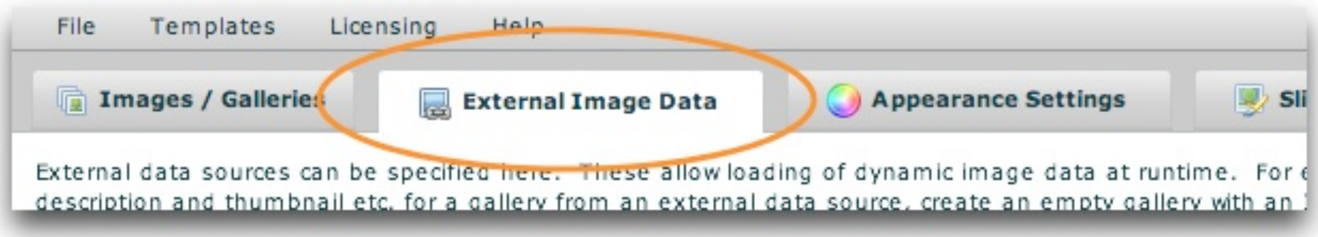
Only add a dynamic data source if you are familiar with their purpose; otherwise, just use the Image Galleries section.

Current External Data Sources:

Gallery Linkage ID	URL

(Click item to edit; drag to re-order)

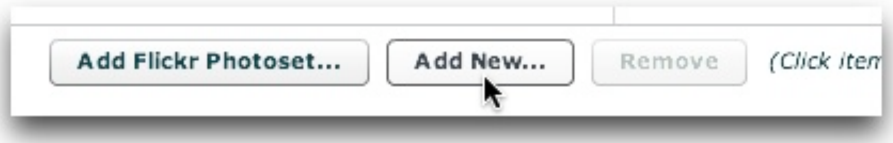
The section is accessed by clicking the 'External Image Data' tab at the top of the wizard window:



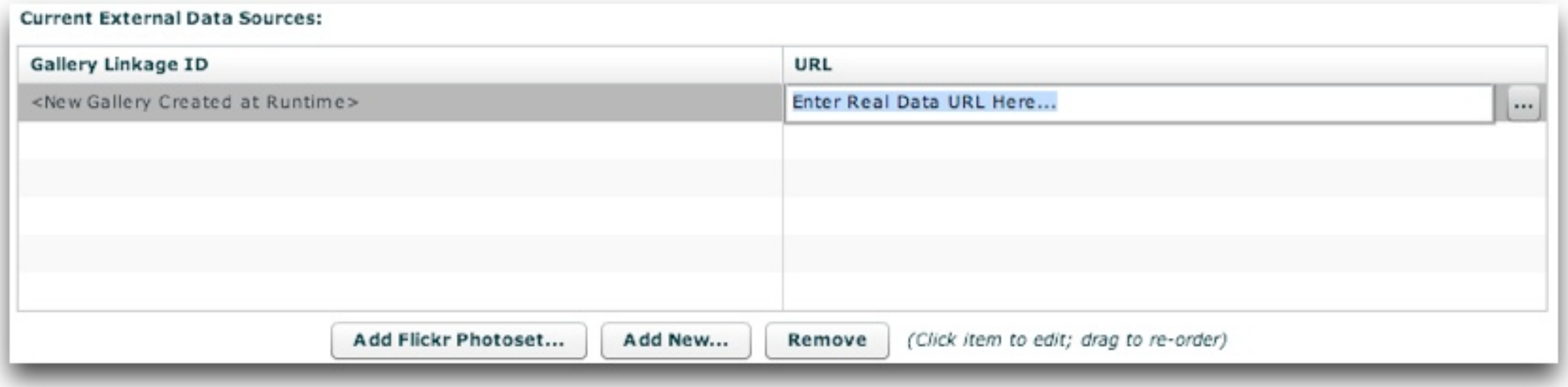
External data sources can be specified here. These allow loading of dynamic image data at runtime. For e
description and thumbnail etc. for a gallery from an external data source, create an empty gallery with an

Adding and Managing External Data Sources

To add a new data source, click the 'Add New...' button:



A new row will appear in the data source grid:

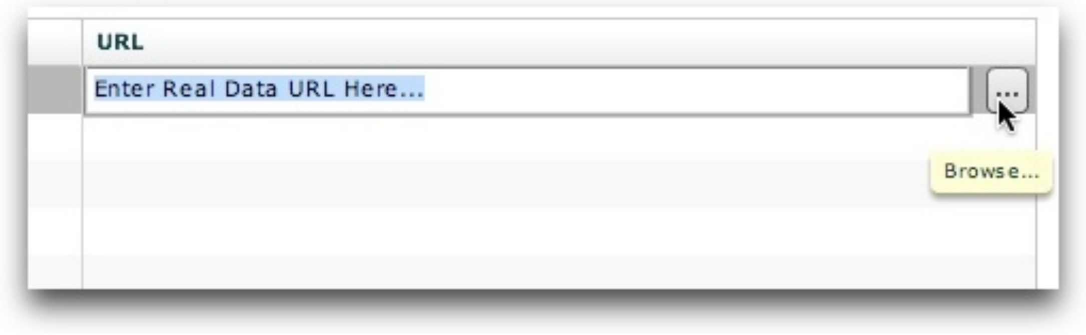


Current External Data Sources:

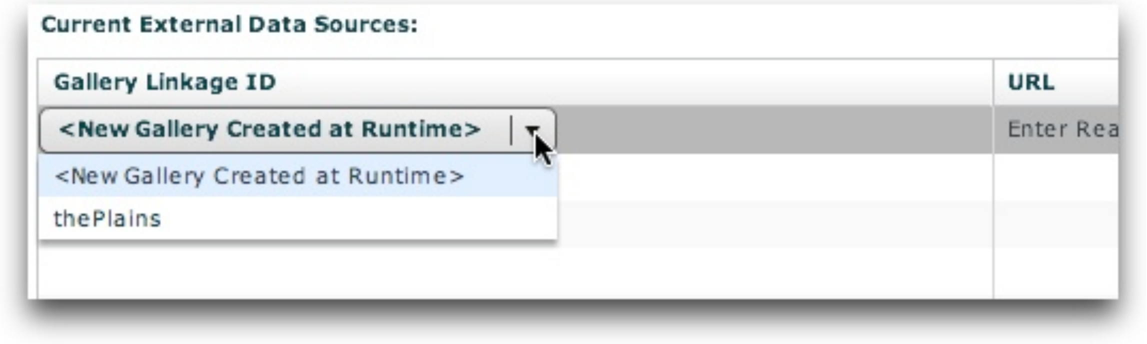
Gallery Linkage ID	URL
<New Gallery Created at Runtime>	Enter Real Data URL Here...

(Click item to edit; drag to re-order)

You now must enter the URL to the dynamic data source. You can either enter the URL directly, or browse to a file by using the browse button on the right edge of the box:



The 'Gallery Linkage ID' setting allows you to specify if the data should be added to a new gallery (created at runtime), or an existing gallery which was defined in the Image Galleries section. To learn more about this, see the Importing Dynamic Data: New Gallery or Existing Gallery section of the Supported Dynamic Data Formats topic. If you have defined galleries with linkage ID values, you can select them by using the dropdown in the grid:


Current External Data Sources:
Gallery Linkage ID
URL
<New Gallery Created at Runtime>
Enter Rea
<New Gallery Created at Runtime>
thePlains

To re-order the dynamic data sources, drag them within the grid. To remove a source, select it then click the Remove button.

Using the Add Flickr Photoset Tool

For more information on using the 'Add Flickr Photoset' tool (a Pro-version-only feature), see the Flickr Integration topic.

Flickr Integration

Yahoo's Flickr image service offers an excellent image management interface. Additionally, Flickr offers several methods by which you can display images from a Flickr account on your own site. This allows for easy integration of slideshows which a client can update as needed. This section of the manual details how to integrate dynamic data from Flickr into your slideshows.

Note: DWUser.com is in no way affiliated with Yahoo! or Flickr. All trademarks are property of their respective owners. This product uses the Flickr API but is not endorsed or certified by Flickr.

Flickr Data Formats

There are two different means by which data can be returned from Flickr: Flickr Feeds, and the Flickr API (with requests for REST data). Here is a quick comparison of the two:

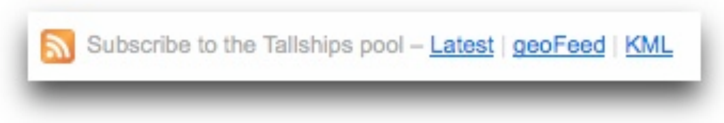
Flickr Feeds: *Pros* - No authentication or key required; publicly available. *Cons* - not much control over the images; limit of only 20 photos; no support for individual photosets.

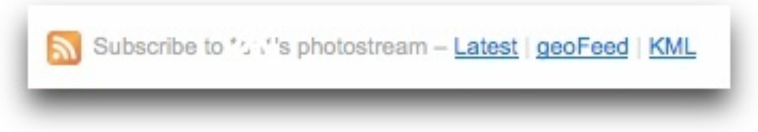
Flickr API: *Pros* - Much more control over image selection, including specific photosets and much more; up to 500 photos in a data request (vs. 20 for feeds). *Cons* - requires registration for a free API key.

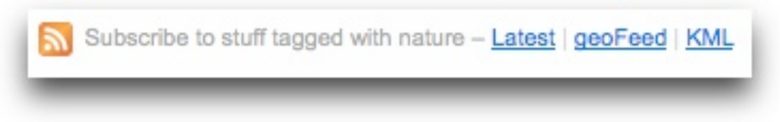
For very simple integrations, the Flickr feeds are generally sufficient. For a more advanced integration -- especially specific photosets or more than 20 images in a gallery -- the Flickr API is the way to go.

Using Flickr Feeds

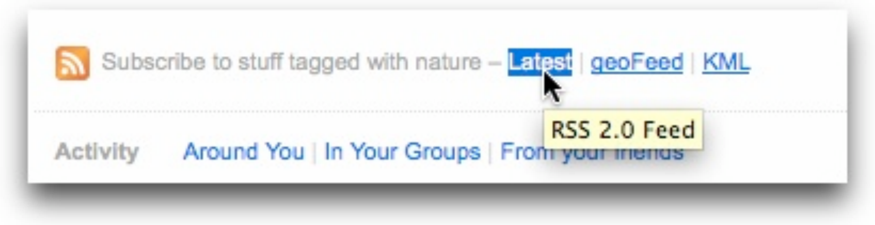
To get a Flickr Feed URL to use, locate a Flickr page which has a feed link at the bottom. For example, link can be found for a group's pool, a user's photostream, or publicly tagged content:


Subscribe to the Tallships pool – Latest | geoFeed | KML


Subscribe to [illegible]'s photostream – Latest | geoFeed | KML


Subscribe to stuff tagged with nature – Latest | geoFeed | KML

On any of these links, click the 'Latest' link. This should take you to the corresponding RSS 2.0 feed:


Subscribe to stuff tagged with nature – Latest | geoFeed | KML

Activity Around You | In Your Groups | From your friends

RSS 2.0 Feed

On the feed page which loads, copy the URL from your browser's address bar. It should look like this:

`http://api.flickr.com/services/feeds/photos_public.gne?tags=nature&lang=en-us&format=rss_200`

Double-check that a `format=...` value exists and that it is set to `format=rss_200`. This will ensure that the feed is in RSS 2.0 format (the format required for integration with a slideshow).

Once you have the Flickr Feed URL, add it directly into the External Data Sources grid.

Using the Flickr API

Important Tip: If you're new to the Flickr API or just want to add a specific photoset, you can use the Add Flickr Photoset tool (usage detailed in section below).

Before loading data from the Flickr API, you must register for a free API key on Flickr's site. Once you have your key, you're ready to proceed.

To learn more about all Flickr API methods, see the Flickr API Overview page. The slideshows can load and integrate data which is returned in the REST format from the Flickr API. **Important:** To be usable, the response must be from one of the methods which returns a <photos> or <photoset> list of photos in the following format:

```
<photos|photoset>
    <photo ... />
    <photo ... />
    ...
</photos|photoset>
```

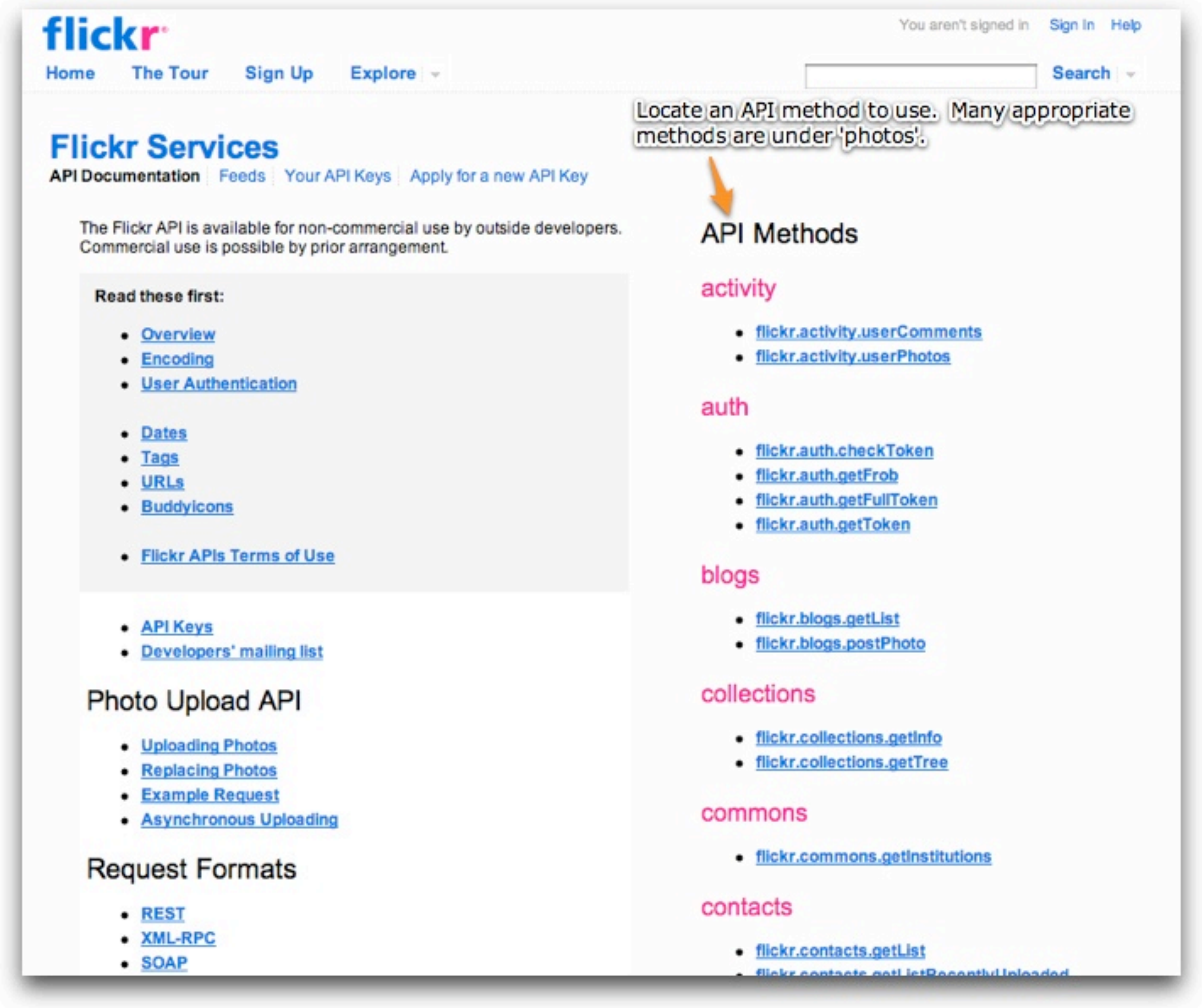
These requests will be made to a URL of the following format:

http://api.flickr.com/services/rest/?method=*MethodName*&*OtherParameters*

You will need to replace *MethodName* with the method name (e.g. flickr.photos.search), and *OtherParameters* with all other parameters, including your API key.

Locating / Building a Flickr API Request URL

The easiest way to build a URL like this is to utilize the Flickr API Explorer. To do this, go to the API homepage and click on the method you want to use on the right. You'll want to do this in **Firefox** because of it's better XML display tools.



Locate an API method to use. Many appropriate methods are under 'photos'.

Flickr Services

API Documentation | Feeds | Your API Keys | Apply for a new API Key

The Flickr API is available for non-commercial use by outside developers. Commercial use is possible by prior arrangement.

Read these first:

- Overview
- Encoding
- User Authentication

- Dates
- Tags
- URLs
- Buddyicons

- Flickr APIs Terms of Use

- API Keys
- Developers' mailing list

Photo Upload API

- Uploading Photos
- Replacing Photos
- Example Request
- Asynchronous Uploading

Request Formats

- REST
- XML-RPC
- SOAP

API Methods

activity

- flickr.activity.userComments
- flickr.activity.userPhotos

auth

- flickr.auth.checkToken
- flickr.auth.getFrob
- flickr.auth.getFullToken
- flickr.auth.getToken

blogs

- flickr.blogs.getList
- flickr.blogs.postPhoto

collections

- flickr.collections.getInfo
- flickr.collections.getTree

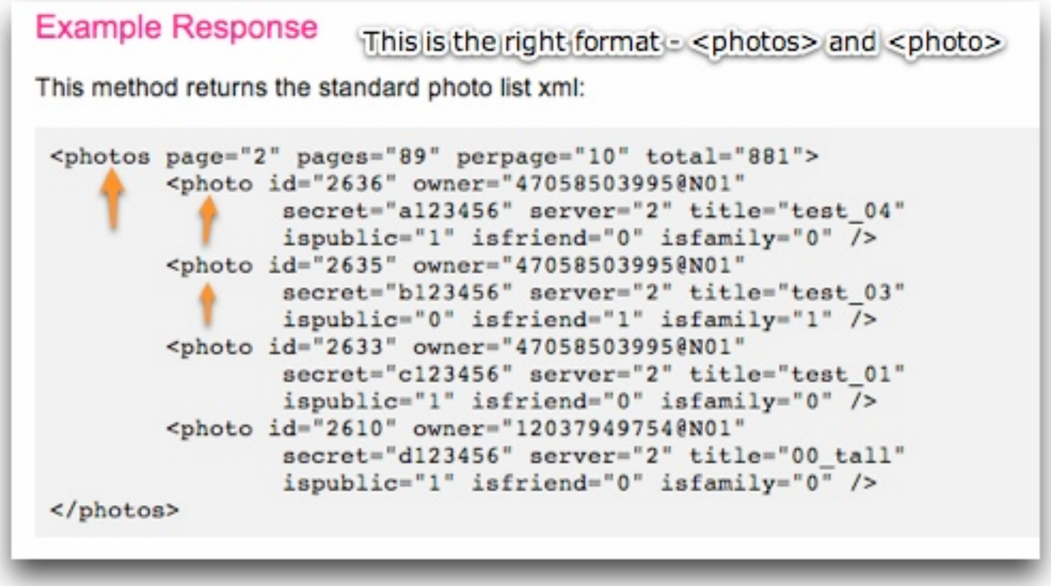
commons

- flickr.commons.getInstitutions

contacts

- flickr.contacts.getList
- flickr.contacts.getListRecentlyUploaded

Most of the appropriate methods are under the 'photos' or 'photosets' headers. In this example, we'll use the flickr.photos.search method, which allows a search of all public photos.   On the method detail page, verify that the response format under 'Example Response' is the <photos> format shown above. This is often referred to as 'photo list xml':



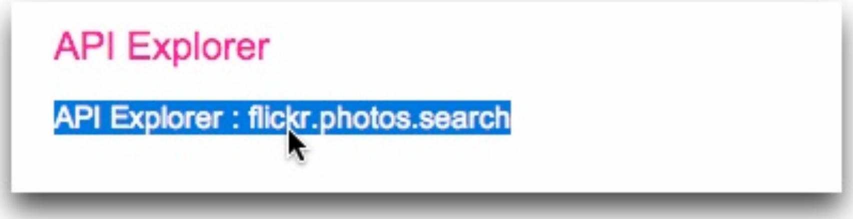
Example Response

This is the right format - <photos> and <photo>

This method returns the standard photo list xml:

```
<photos page="2" pages="89" perpage="10" total="881">
        <photo id="2636" owner="47058503995@N01"
                secret="a123456" server="2" title="test_04"
                ispublic="1" isfriend="0" isfamily="0" />
        <photo id="2635" owner="47058503995@N01"
                secret="b123456" server="2" title="test_03"
                ispublic="0" isfriend="1" isfamily="1" />
        <photo id="2633" owner="47058503995@N01"
                secret="c123456" server="2" title="test_01"
                ispublic="1" isfriend="0" isfamily="0" />
        <photo id="2610" owner="12037949754@N01"
                secret="d123456" server="2" title="00_tall"
                ispublic="1" isfriend="0" isfamily="0" />
</photos>
```

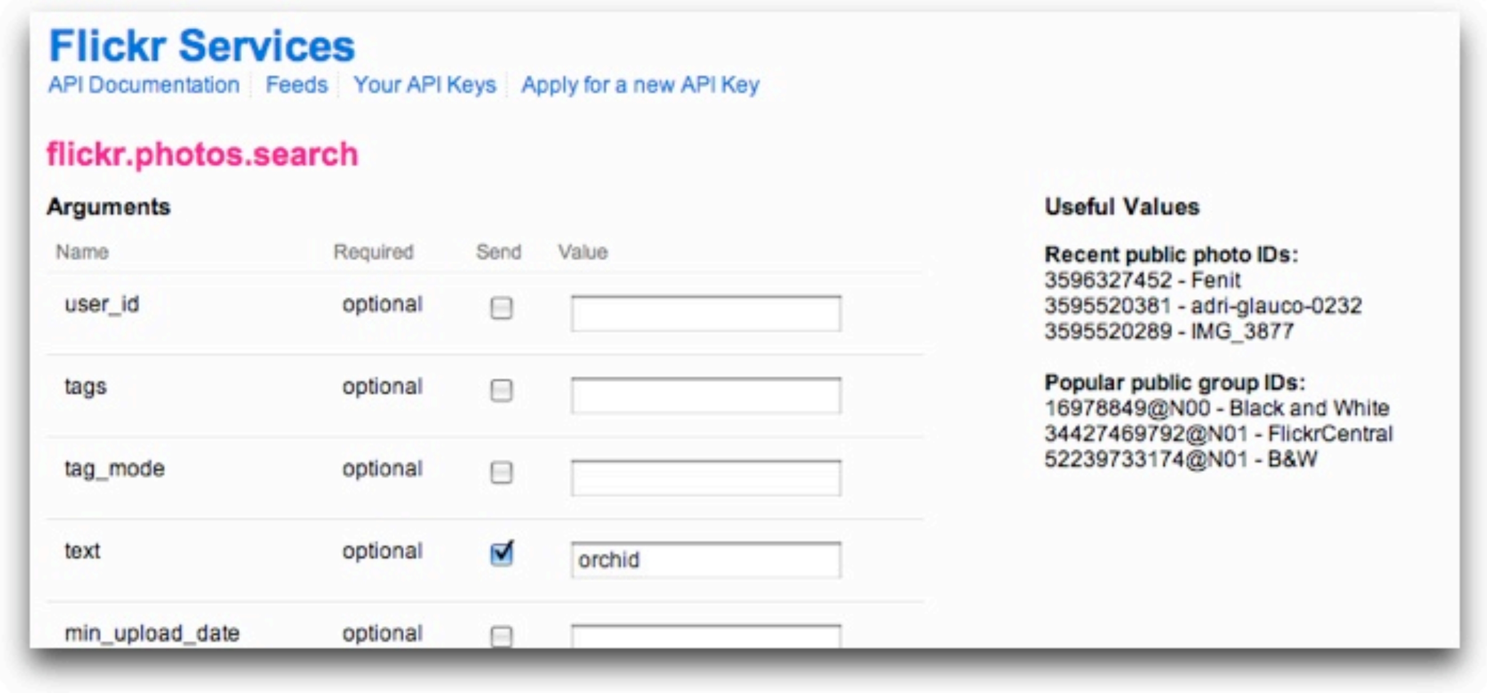
Then, scroll to the bottom of the page and click the API Explorer link:



API Explorer

API Explorer : flickr.photos.search

On the API Explorer page, you can enter all of the relevant parameters (except for your API key).   For this example, only the **text** parameter needs to be specified.



flickr.photos.search

Arguments

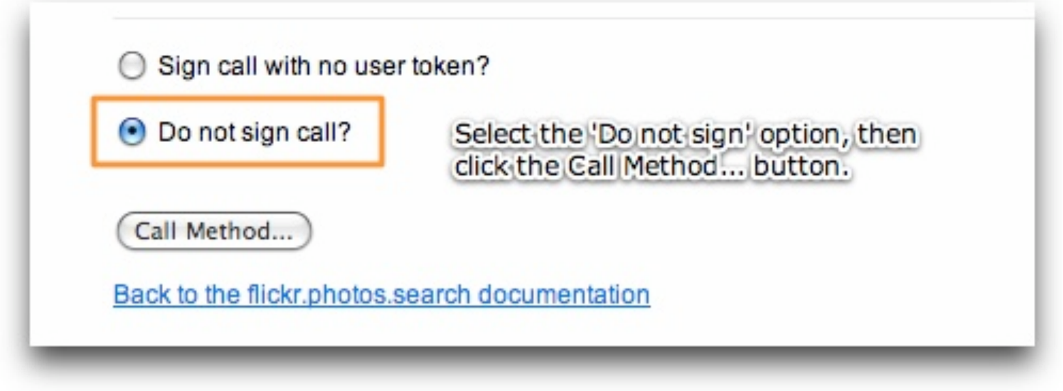
Name	Required	Send	Value
user_id	optional	☐	
tags	optional	☐	
tag_mode	optional	☐	
text	optional	☑	orchid
min_upload_date	optional	☐	

Useful Values

Recent public photo IDs:
3596327452 - Fenit
3595520381 - adri-glauco-0232
3595520289 - IMG_3877

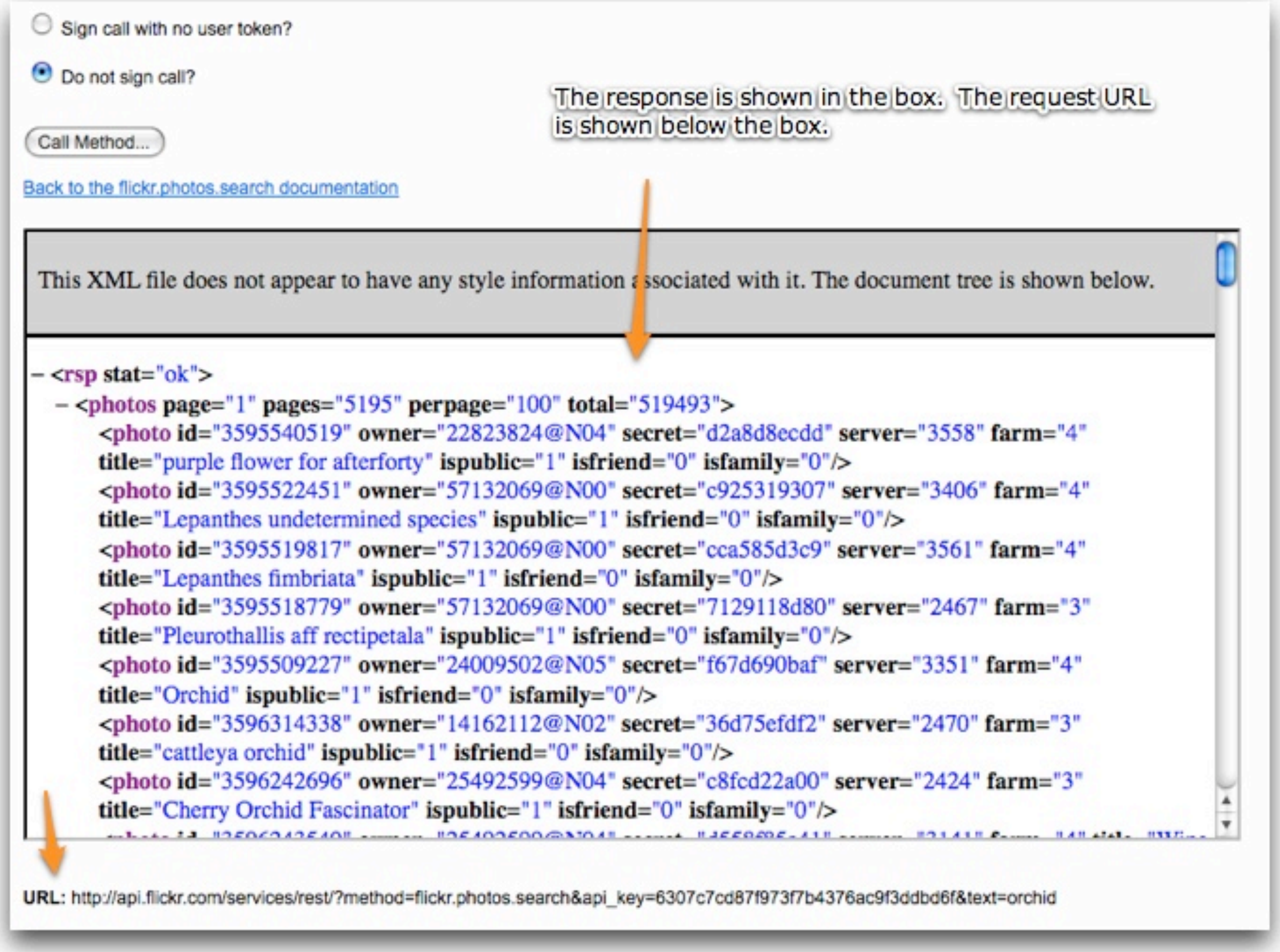
Popular public group IDs:
16978849@N00 - Black and White
34427469792@N01 - FlickrCentral
52239733174@N01 - B&W

Scroll down to the bottom, select the 'Do Not Sign Call' option and click Call Method:


○ Sign call with no user token?
● Do not sign call?
Call Method...
Back to the flickr.photos.search documentation

Select the 'Do not sign' option, then click the Call Method... button.

The page will reload. Scroll to the bottom, and you'll see the response of your request:



○ Sign call with no user token?

● Do not sign call?

The response is shown in the box. The request URL is shown below the box.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
– <rsp stat="ok">
  – <photos page="1" pages="5195" perpage="100" total="519493">
      <photo id="3595540519" owner="22823824@N04" secret="d2a8d8ecdd" server="3558" farm="4"
      title="purple flower for afterforty" ispublic="1" isfriend="0" isfamily="0"/>
      <photo id="3595522451" owner="57132069@N00" secret="c925319307" server="3406" farm="4"
      title="Lepanthes undetermined species" ispublic="1" isfriend="0" isfamily="0"/>
      <photo id="3595519817" owner="57132069@N00" secret="cca585d3c9" server="3561" farm="4"
      title="Lepanthes fimbriata" ispublic="1" isfriend="0" isfamily="0"/>
      <photo id="3595518779" owner="57132069@N00" secret="7129118d80" server="2467" farm="3"
      title="Pleurothallis aff rectipetala" ispublic="1" isfriend="0" isfamily="0"/>
      <photo id="3595509227" owner="24009502@N05" secret="f67d690baf" server="3351" farm="4"
      title="Orchid" ispublic="1" isfriend="0" isfamily="0"/>
      <photo id="3596314338" owner="14162112@N02" secret="36d75efdf2" server="2470" farm="3"
      title="cattleya orchid" ispublic="1" isfriend="0" isfamily="0"/>
      <photo id="3596242696" owner="25492599@N04" secret="c8fcd22a00" server="2424" farm="3"
      title="Cherry Orchid Fascinator" ispublic="1" isfriend="0" isfamily="0"/>
```

URL: http://api.flickr.com/services/rest/?method=flickr.photos.search&api_key=6307c7cd87f973f7b4376ac9f3ddbd6f&text=orchid

Below the response box, the URL for the request will be displayed. This is the URL which you'll use as the dynamic data source URL. **Important:** Before using this URL, you must substitute in your API key. If you don't, the call will cease to function after a few hours because the default key is only a temporary key which expires. To do this, locate this portion of the URL:

`...api_key=`***`gibberish`***`&...`

You'll need to replace the gibberish between `api_key=` and the ampersand with your API key. Next, if desired, you need to edit the URL to allow for descriptions (captions) to come through with your photos and the titles. To do this, add ***&extras=description*** at the end of the URL:

```
...&extras=description
```

Finally, if you want to add support for displaying videos, you need to modify the **extras** value to include **media** like this:

...&extras=media or **...&extras=description,media**

You can now paste this URL into the URL field in the External Data Sources section of the wizard.

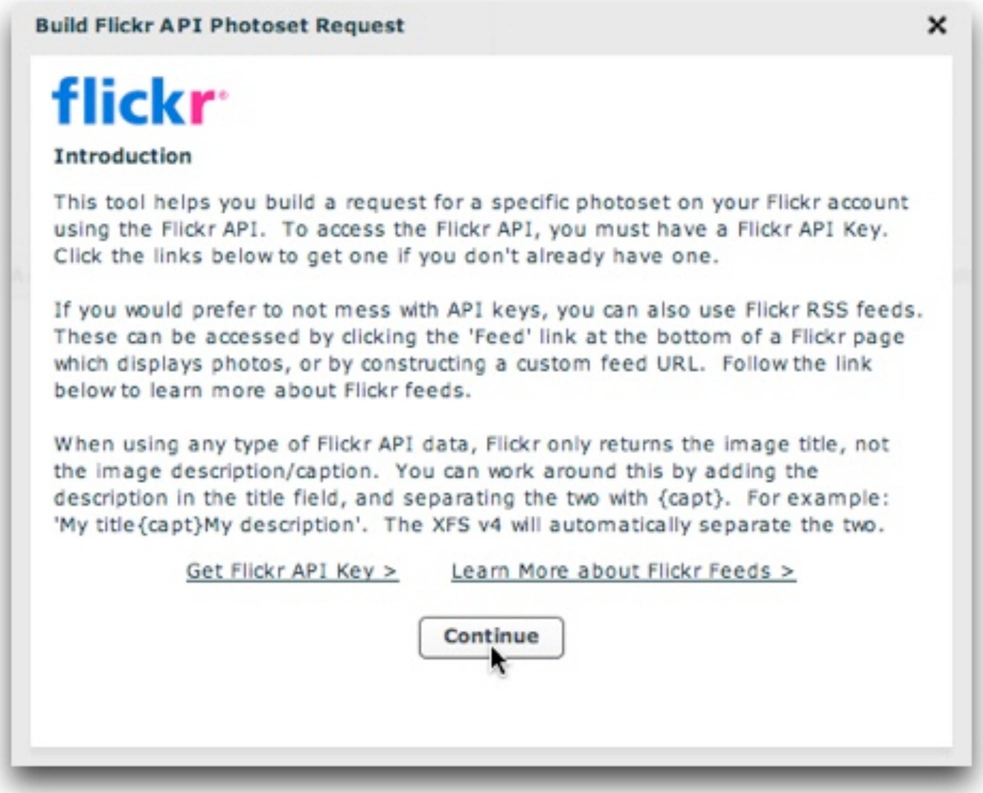
Using the Add Flickr Photoset Tool (Pro Version Only)

The Add Flickr Photoset tool allows you to quickly and easily build a reference to a specific photoset in your Flickr account without knowing anything about Flickr API requests. This feature is only available in the Pro version; to learn about upgrading, click here.

Before loading data from the Flickr API, you must register for a free API key on Flickr's site. Once you have your key, you're ready to proceed.

In the main External Image Data section, click the 'Add Flickr Photoset...' button:

This will open the Add Flickr Photoset dialog, which appears as follows:



Introduction

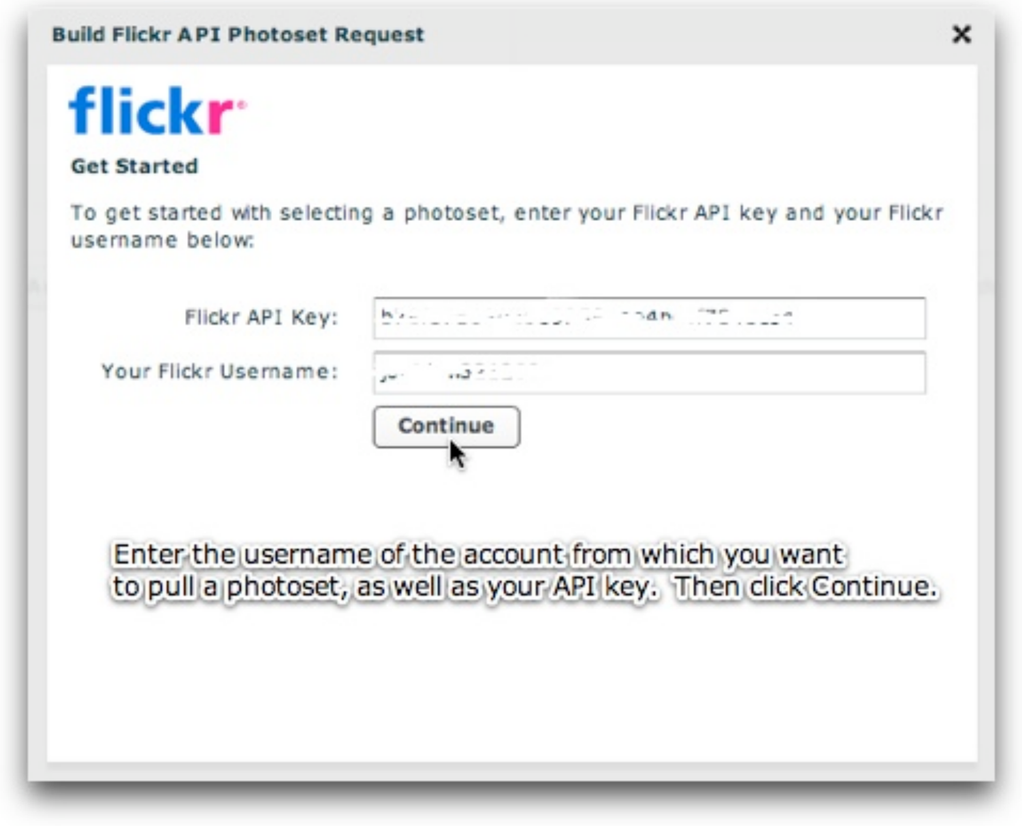
This tool helps you build a request for a specific photoset on your Flickr account using the Flickr API. To access the Flickr API, you must have a Flickr API Key. Click the links below to get one if you don't already have one.

If you would prefer to not mess with API keys, you can also use Flickr RSS feeds. These can be accessed by clicking the 'Feed' link at the bottom of a Flickr page which displays photos, or by constructing a custom feed URL. Follow the link below to learn more about Flickr feeds.

When using any type of Flickr API data, Flickr only returns the image title, not the image description/caption. You can work around this by adding the description in the title field, and separating the two with {capt}. For example: 'My title{capt}My description'. The XFS v4 will automatically separate the two.

As noted in the dialog, when using any type of Flickr API data, Flickr only returns the image title, not the image description/caption. You can work around this by adding the description in the title field, and separating the two with `{desc}`. To learn more about format transformations, see this page.

Once you click Continue, you will be prompted to enter the username of the Flickr account which holds the photoset(s) you want to use, and your Flickr API key:



Build Flickr API Photoset Request

Get Started

To get started with selecting a photoset, enter your Flickr API key and your Flickr username below:

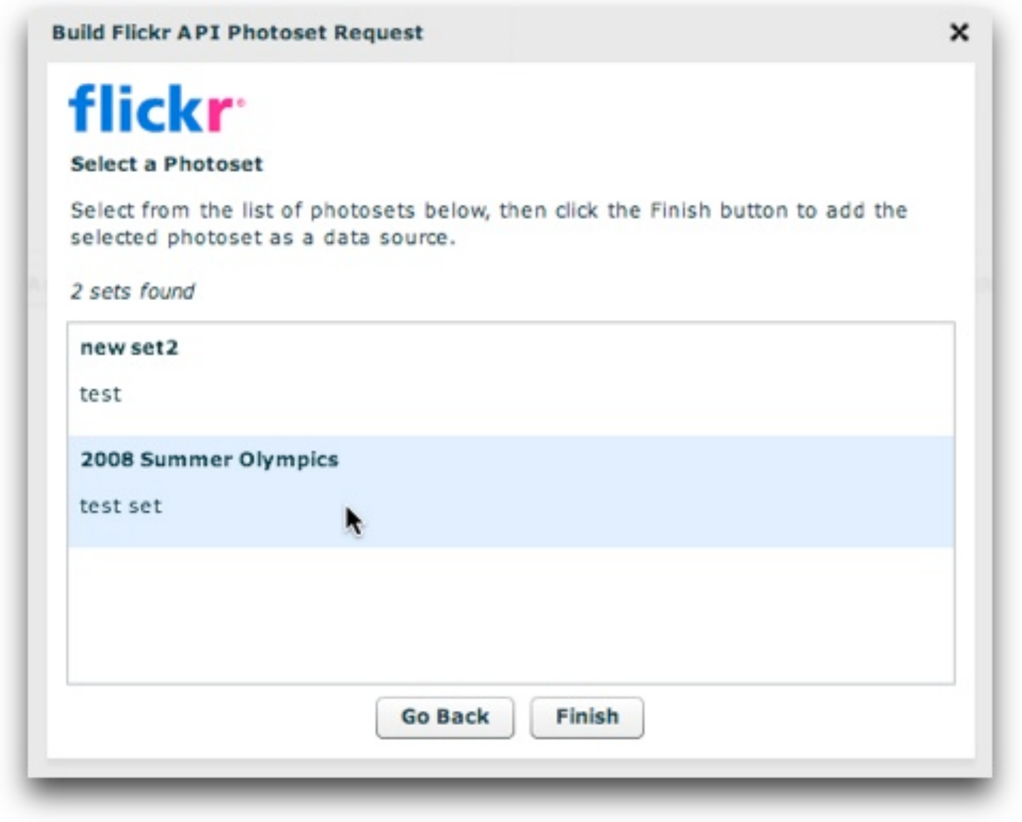
Flickr API Key: [illegible]

Your Flickr Username: [illegible]

Enter the username of the account from which you want to pull a photoset, as well as your API key. Then click Continue.

Please note: You should only continue in this portion of the wizard if you're using the Flickr API (*not* Flickr feeds). If you're using a Flickr feed (as described above), you'll add the URL directly to the External Data Sources grid.

Click Continue, and the list of available photosets will be loaded. In the list, select the photoset you want to use:



Build Flickr API Photoset Request

Select a Photoset

Select from the list of photosets below, then click the Finish button to add the selected photoset as a data source.

2 sets found

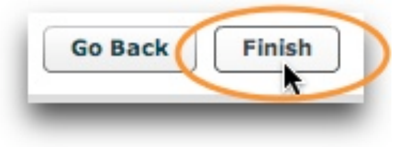
new set2

test

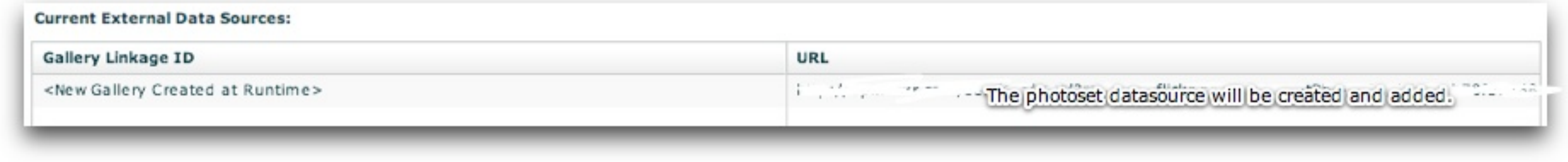
2008 Summer Olympics

test set

If you want to load from a different account, click the Go Back button.   Otherwise, click the Finish button to exit the window and add the photoset:



The source will be added to the External Data Source grid.   If you want to add the photo data to a specific gallery, specify a value for the Gallery Linkage ID option.   To learn more about this, see the Importing Dynamic Data: New Gallery or Existing Gallery section of the Supported Dynamic Data Formats topic.



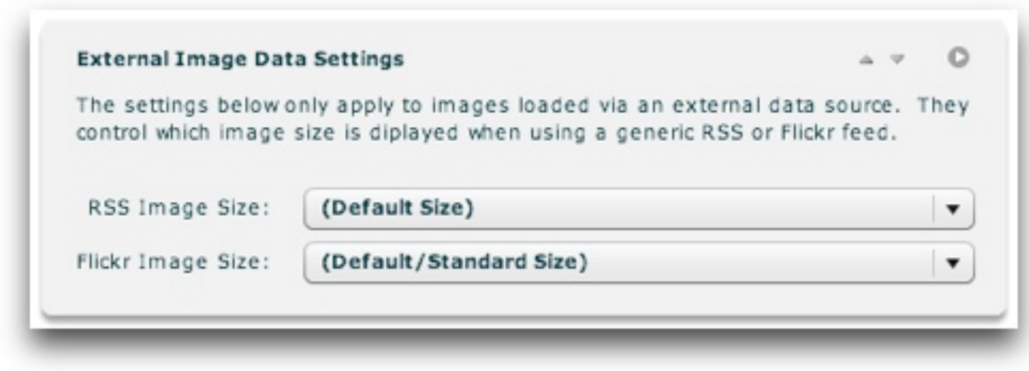
Current External Data Sources:

Gallery Linkage ID	URL
<New Gallery Created at Runtime>	The photoset datasource will be created and added.

When you view the slideshow, the photoset will be loaded and displayed from Flickr.

Specifying the Flickr Image Sizes

When using a Flickr feed or Flickr API data, there are multiple image sizes available to choose from. The size you choose will depend on the size of your slideshow. You can specify the default size in the main 'Global Configuration and Playback Settings' tab. Expand the 'External Image Data Settings' pod, and change the default 'Flickr Image Size' as desired:



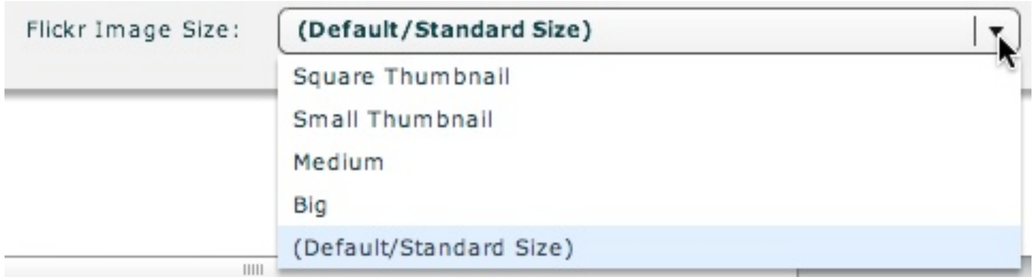
External Image Data Settings

The settings below only apply to images loaded via an external data source. They control which image size is diplayed when using a generic RSS or Flickr feed.

RSS Image Size: (Default Size)

Flickr Image Size: (Default/Standard Size)

There are several options; an explanation is shown in the following table:



Flickr Image Size: (Default/Standard Size)

Size	Description
Square Thumbnail	Small square thumbnail, 75x75 pixels; image is cropped.
Small Thumbnail	Small scaled version of original, 100 pixels on longest side; aspect ratio is maintained.
Medium	Medium-size scaled version of original, 240 pixels on longest side.
(Default / Standard Size)	The default (standard) size; 500 pixels on longest side.
Big	Large, 1024 pixels on longest side. Only exists for images with originals this large.

For more information about the Global Configuration and Playback Settings area of the wizard, see the Using the 'Global Configuration and Playback Settings' Section portion of the manual.

Default Flickr Image Links

To comply with the Flickr TOS, all photos retrieved from Flickr will by default link to their associated Flickr page when clicked on. You can disable this by manually making a change in the slideshow's XML configuration file. Locate the main `<galleries>` node and add the following attribute:

```
<galleries flickrIgnoreLink="true">
```

This setting applies to RSS feeds from other sources as well. You can learn more about the XML configuration file format and how to use it by viewing the Developer Reference.

Please note: *DWUser.com is in no way responsible for your compliance with (or failure to comply with) the Flickr TOS. You are solely responsible for understanding and abiding by all terms and usage guidelines established by Flickr or Yahoo.*

Creating Data Sources + Example Code

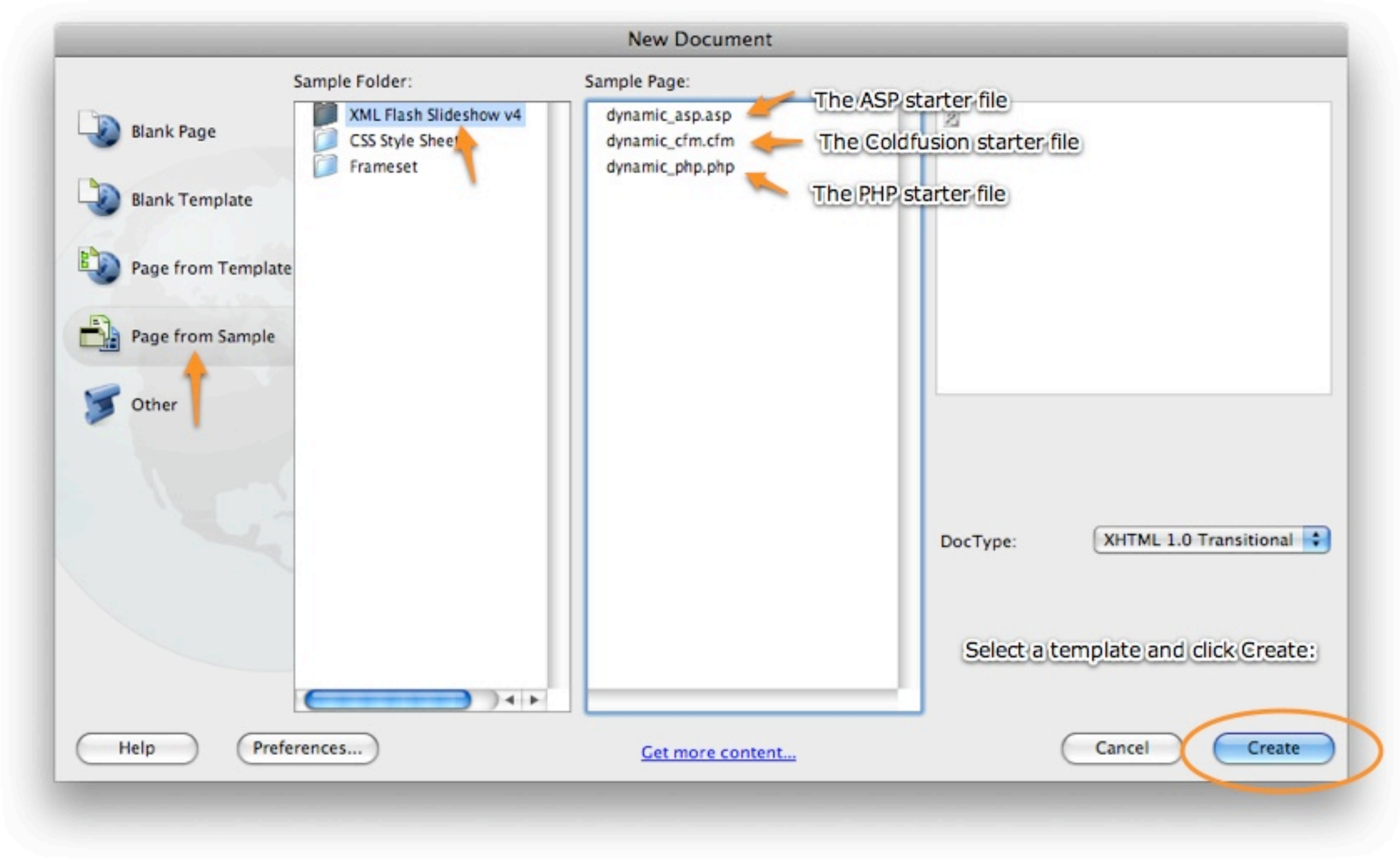
One of the most powerful applications of dynamic data is creating your own data sources. This section gives an overview of this process and includes some example code for common applications.

Creating your own dynamic data source files requires two things:

- Server support for a scripting language, such as PHP (which is used for the examples in the tutorial), ASP, Coldfusion, .NET, JSP, etc.
- Familiarity with server-side programming in this language. If you're not familiar with this type of programming, it is recommended that you don't attempt this type of integration.
- Rudimentary familiarity with XML

Creating the Dynamic Data Source File

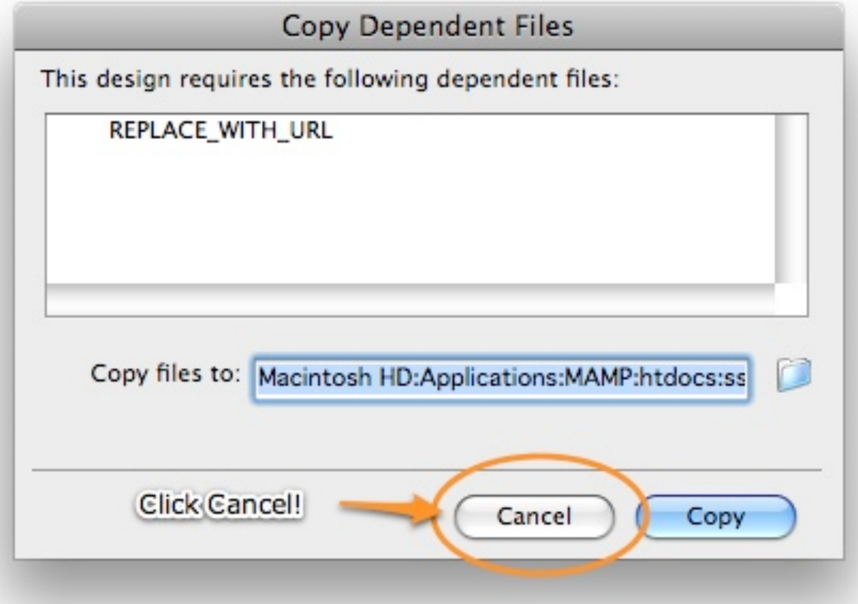
Included with the extension are several dynamic-data-source starter files, which provide a starting point for PHP, ASP, and Coldfusion data sources. To access these templates, open the New Document dialog (File > New...), and select the Page From Sample > XML Flash Slideshow v4 folder. The appearance of the dialog will vary slightly in older versions of Dreamweaver, but will be similar:



The `dynamic_asp.asp` file is for ASP, `dynamic_cfm.cfm` for Coldfusion, and `dynamic_php.php` for PHP implementations.

Important: Please note that these templates are for dynamic data sources, and are NOT pages into which slideshows should be inserted! Slideshows must be added to standard HTML documents as shown in earlier portions of the manual.

Once you click Create, you may receive the following prompt:



Copy Dependent Files

This design requires the following dependent files:

REPLACE_WITH_URL

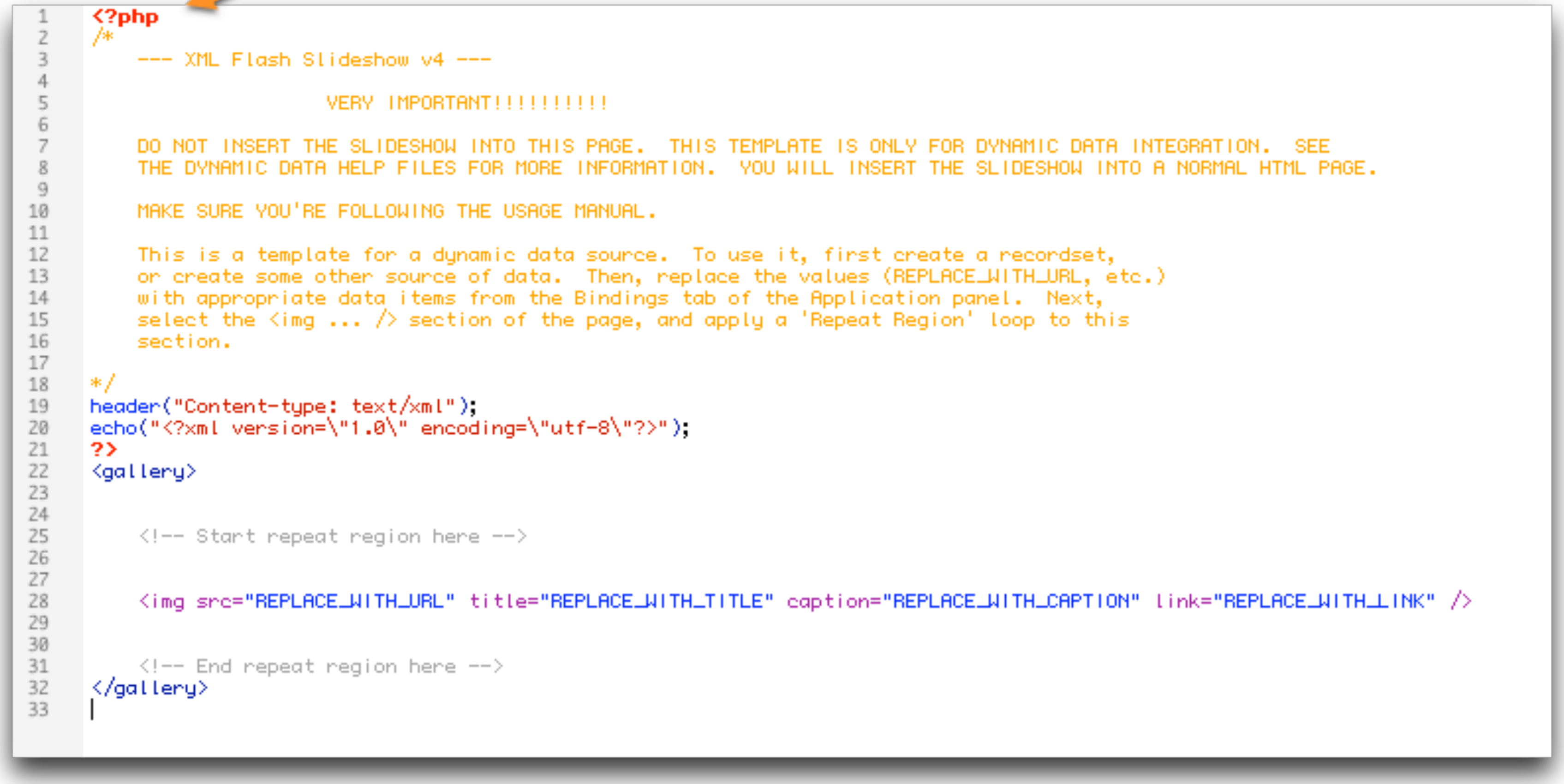
Copy files to: Macintosh HD:Applications:MAMP:htdocs:ss

This is due to Dreamweaver thinking that the example code is requesting a dependent file to be copied. This is incorrect, and so **Cancel** should be clicked.

After creating the new document, switch to Code View. Sometimes Dreamweaver will add extraneous code at the top of the file, such as `<!-- DW6 -->`. This code must be removed, so that there is no code prior to the opening server-side code block (e.g. `<?php ...`).

If you're using PHP, the blank template file should look like this:

All extraneous code at the top of the file (such as <!--DW6-->) was removed.



```
<?php
/*
    --- XML Flash Slideshow v4 ---

                    VERY IMPORTANT!!!!!!!!!!

    DO NOT INSERT THE SLIDESHOW INTO THIS PAGE.  THIS TEMPLATE IS ONLY FOR DYNAMIC DATA INTEGRATION.  SEE
    THE DYNAMIC DATA HELP FILES FOR MORE INFORMATION.  YOU WILL INSERT THE SLIDESHOW INTO A NORMAL HTML PAGE.

    MAKE SURE YOU'RE FOLLOWING THE USAGE MANUAL.

    This is a template for a dynamic data source.  To use it, first create a recordset,
    or create some other source of data.  Then, replace the values (REPLACE_WITH_URL, etc.)
    with appropriate data items from the Bindings tab of the Application panel.  Next,
    select the <img ... /> section of the page, and apply a 'Repeat Region' loop to this
    section.

*/
header("Content-type: text/xml");
echo("<?xml version=\"1.0\" encoding=\"utf-8\"?>");
?>
<gallery>


    <!-- Start repeat region here -->


    <img src="REPLACE_WITH_URL" title="REPLACE_WITH_TITLE" caption="REPLACE_WITH_CAPTION" link="REPLACE_WITH_LINK" />


    <!-- End repeat region here -->
</gallery>

```

Save the document in your site.   It is usually best to save it in the same folder as the HTML page holding your slideshow (to simplify path creation).

The Data Source's XML Format

The file can use either the Single Gallery Format (shown in this tutorial) or the Multiple Galleries Format (requires more advanced programming not covered in this tutorial).   Both of these formats are described in the Supported Dynamic Data Formats topic, with the Single Gallery Format detailed further in the Developer Reference.

When your file is finished, you should be able to preview the file in a browser, and see a properly-formed XML document. The format of this XML is very important, to ensure that the XML can be properly read by your slideshow. The end result must be in the following format:

```
<gallery>
    <img src="images/image1.jpg" title="Image 1 title" caption="" link="" />
    <img src="images/image2.jpg" title="Image 2 title" caption="" link="" />
    <img src="images/image3.jpg" title="Image 3 title" caption="" link="" />
</gallery>
```

There is a <gallery> node which holds multiple <img /> nodes. Each <img /> node represents a single image. Each attribute specifies the following:

- `src="..."` portion represents the URL of the JPEG image (required)
- `thumb="..."` portion represents the URL of the thumbnail image (optional)
- `title="..."` portion represents the title for the image (optional)
- `caption="..."` portion represents the caption for the image (optional)
- `link="..."` portion represents the hyperlink for the image (optional)

Integrating the Dynamic Data

Now that you have your blank template, you're ready to bring it to life with some dynamic data. Do do this, you must start with a dynamic list of images. This can come from many different sources, including a database recordset (most common), an array of files in a directory, or a static list of images that you have randomized. More details about these different sources will be mentioned later on.

To translate this list into the proper XML format, you must create a loop in the dynamic data document. This loop will go over the list of images, outputting an `<img src="..." />` node for each of these items.

The appropriate locations for the beginning and end of this loop are denoted by `<!--Start repeat region here-->` and `<!--End repeat region here-->` in the dynamic data file template.

Important note about image paths: Regardless of where the dynamic data source file is located, the image paths must be relative to the HTML page holding the slideshow. Failure to do this will result in 'Image failed to load' messages.

In the following sections of this topic, there are specific examples (with code) of various ways to get and use a list of images:

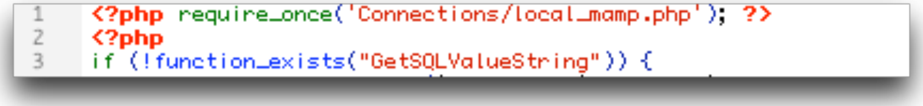
- Using a Database Recordset
- Dynamically Listing All Images in a Directory on the Server

Specific Example: Using a Database Recordset

In this example, integration of image data from a database will be shown. While this example is shown for PHP, the techniques are fully applicable to other server languages as well.

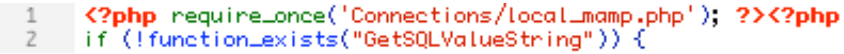
First, you must use the Application panel to insert a recordset into the file. If you are not familiar with creating a Database connection and a Recordset, you will need to first familiarize yourself with basic Dreamweaver dynamic data concepts. To learn about this type of Dreamweaver development, see these pages in the Dreamweaver docs: Preparing to Build Dynamic Sites and Making Pages Dynamic.

Make sure that the recordset code doesn't include any linebreaks or other whitespace between code blocks. For example:



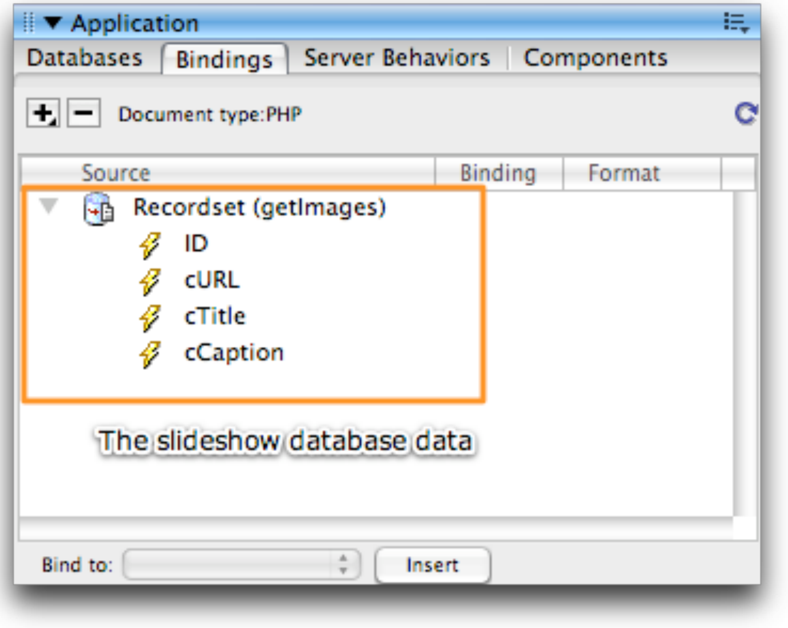
```
<?php require_once('Connections/local_mamp.php'); ?>
<?php
if (!function_exists("GetSQLValueString")) {
```

Must be changed to (by removing the linebreak):



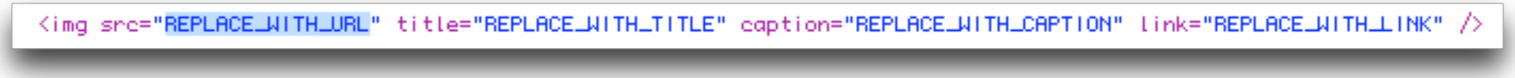
```
<?php require_once('Connections/local_mamp.php'); ?><?php
if (!function_exists("GetSQLValueString")) {
```

Now, switch to the Bindings section of the Application panel (this example shown in Dreamweaver CS3; other versions are similar). Expand the recordset so that you can see each of the recordset fields:



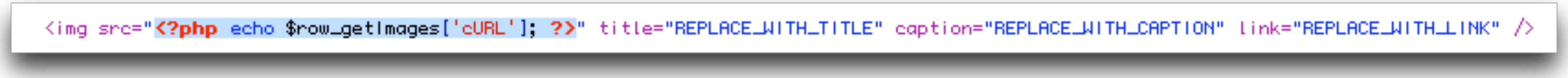
The slideshow database data

In this example, the recordset has been named `getImages`. The database table has three relevant columns: `cURL` (the URL to the image file to be used), `cTitle` (the title associated with the image), and `cCaption` (the caption/description associated with the image). In the document's code, locate the `<img ... />` line (which represents the image), and select the `src` attribute:



```
<img src="REPLACE_WITH_URL" title="REPLACE_WITH_TITLE" caption="REPLACE_WITH_CAPTION" link="REPLACE_WITH_LINK" />
```

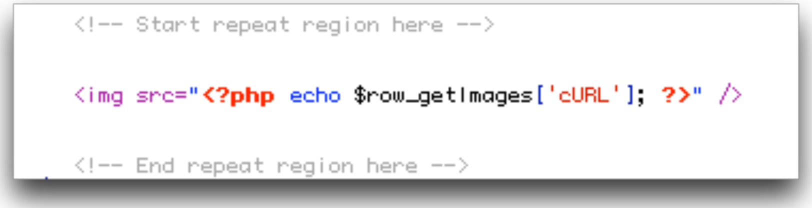
Back in the Bindings panel, select the recordset field which represents the image URL (`cURL` in this example), and drag it onto the `REPLACE_WITH_URL` selection. It should replace the current contents, leaving you with something like this:



```
<img src="<?php echo $row_getImages['cURL']; ?>" title="REPLACE_WITH_TITLE" caption="REPLACE_WITH_CAPTION" link="REPLACE_WITH_LINK" />
```

If you want to use a title, caption, link, or thumbnail from the database, replace the other attributes with data from the Bindings panel. If you don't plan to use the other attributes, simply remove them.

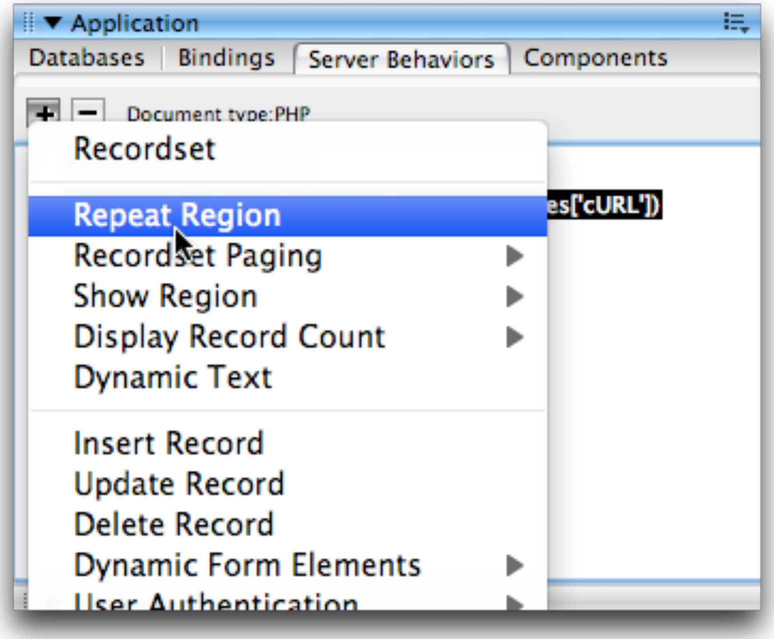
The `<img ... />` section of your document should now look like this:



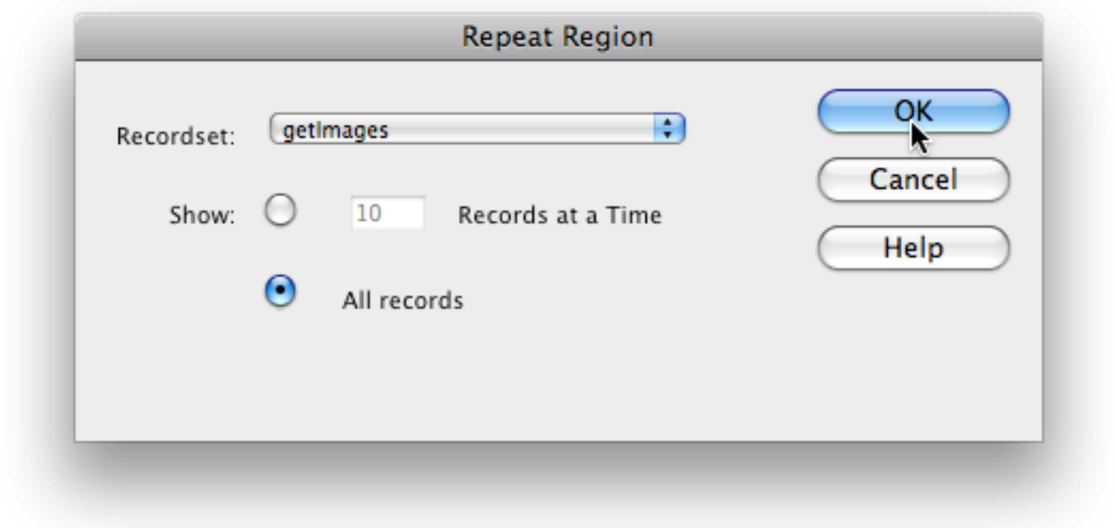
```
<!-- Start repeat region here -->

<img src="<?php echo $row_getImages['cURL']; ?>" />

<!-- End repeat region here -->
```

Note: If your title or caption values contain special characters (e.g. a quote or ampersand), you need to modify the PHP code to include the `htmlspecialchars` function. For example, you might have: `<?php echo( htmlspecialchars($row_getImages['cCaption']) ); ?>`

You are now ready to setup the looping for the `<img ... />` node. Select the entire `<img ... />` line, and switch to the Server Behaviors section of the Application Panel. From the add behavior menu, select the Repeat Region option:



In the Repeat Region dialog, select the images recordset (`getImages` in this example), select the 'All Records' option, and click OK:

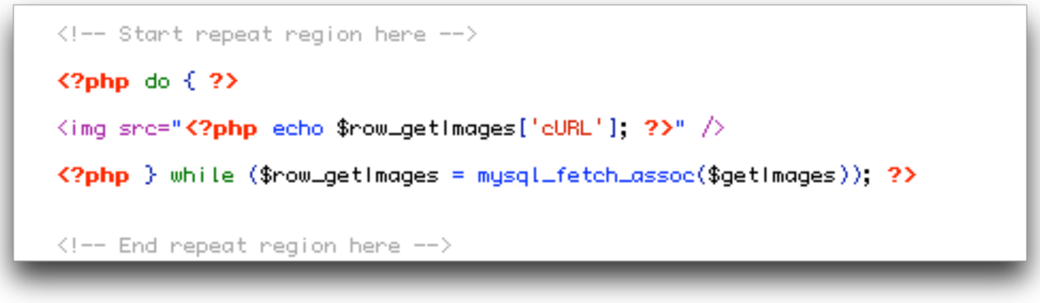


Recordset: getImages

Show: ○ 10 Records at a Time

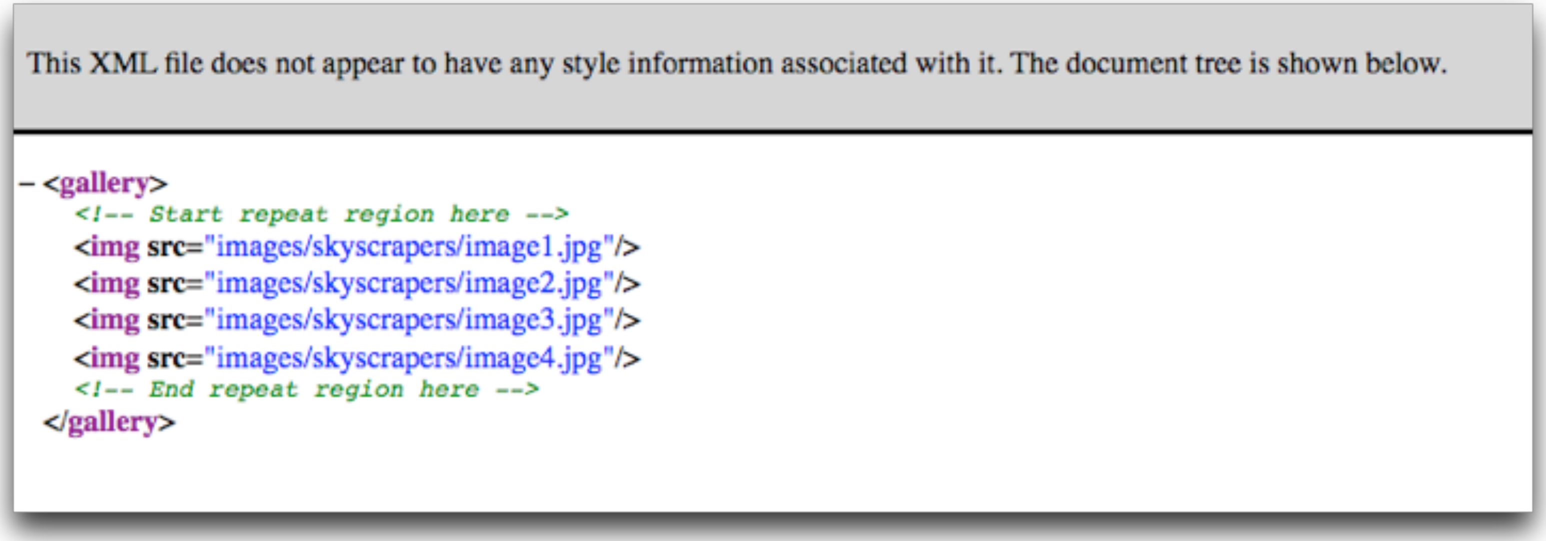
● All records

The repeater code will be added to your document. The `<img ... />` portion should now look similar to this:



```
<!-- Start repeat region here -->
<?php do { ?>
<img src="<?php echo $row_getImages['cURL']; ?>" />
<?php } while ($row_getImages = mysql_fetch_assoc($getImages)); ?>

<!-- End repeat region here -->
```

This loop will output an `<img ... />` node for each record in the database recordset. You are now ready to preview the file in a browser and verify that everything is properly configured. Upload the page if necessary, then open it in a browser. Firefox is recommended because of its superior XML viewing tools. You should see something like this:



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
– <gallery>
    <!-- Start repeat region here -->
    <img src="images/skyscrapers/image1.jpg"/>
    <img src="images/skyscrapers/image2.jpg"/>
    <img src="images/skyscrapers/image3.jpg"/>
    <img src="images/skyscrapers/image4.jpg"/>
    <!-- End repeat region here -->
  </gallery>
```

As intended, this is a dynamically-generated list of images pulled from the database. If you instead see a PHP error about *'Output already started at line N...'*, then make sure that you have no whitespace or other extraneous code before the very first `<?php` code block. Additionally, make sure that there aren't any linebreaks or other whitespace characters between the first couple of `<?php ... ?>` blocks.

Once everything appears correctly, the dynamic data file is ready to be added in the wizard as a source of external image data.

Specific Example: Dynamically Listing All Images in a Directory on the Server

To use all images in a directory, you first need to get the list of all images. To begin, add the following PHP code block directly above the `<gallery>` node in the blank file:

```
<?php
// Specify the path to the images directory on the server, **relative to this file**
$dir = "images/skyscrapers/";

// Specify the path to the images directory, **relative to the slideshow's HTML page**
$relativePathTo = "images/skyscrapers/";

// Create an holder array for the image names
$imageNames = array();
```

```
// Get the list of images
if ($dh = opendir($dir))
{
        while (($file = readdir($dh)) !== false)
        {
                // Make sure that this is a JPEG image.  You can change this expression if you are using PNGs, GIFs, or SWFs
                // For example: if (preg_match("|.png$|i", $file) == false)
                if (preg_match("|.jpe?g$|i", $file) == false)
                {
                        continue;
                }

                // Add the image to the image array
                $imageNames[] = $file;
        }
        closedir($dh);
}
?>
```

To work properly, you must update these two lines to point to the images directory:

```
// Specify the path to the images directory on the server, **relative to this file**
$dir = "images/skyscrapers/";

// Specify the path to the images directory, **relative to the slideshow's HTML page**
$relativePathTo = "images/skyscrapers/";
```

As stated in the comments, the first reference must be relative to the data source file and the second relative to the slideshow's HTML page.   In this instance, this is the same because the data source file is in the same directory as the slideshow's HTML page.   The code will get all of the jpeg images in the directory and place them in the `$imageNames` array.

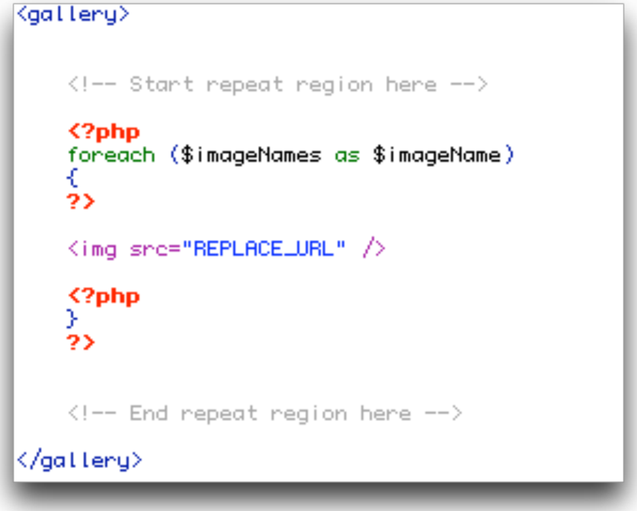
Next, looping code must be added.   Above the `<img ... />` node, add the following:

```
<?php
foreach ($imageNames as $imageName)
{
?>
```

Directly below the <img ... /> node, add this:

```
<?php
}
?>
```

Your code should now look like this:

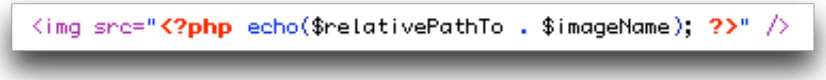


```
<gallery>


    <!-- Start repeat region here -->

    <?php
    foreach ($imageNames as $imageName)
    {
    ?>

    <img src="REPLACE_URL" />

    <?php
    }
    ?>


    <!-- End repeat region here -->

</gallery>
```

Now that the loop is set up, replace the `REPLACE_WITH_URL` text with the following code:

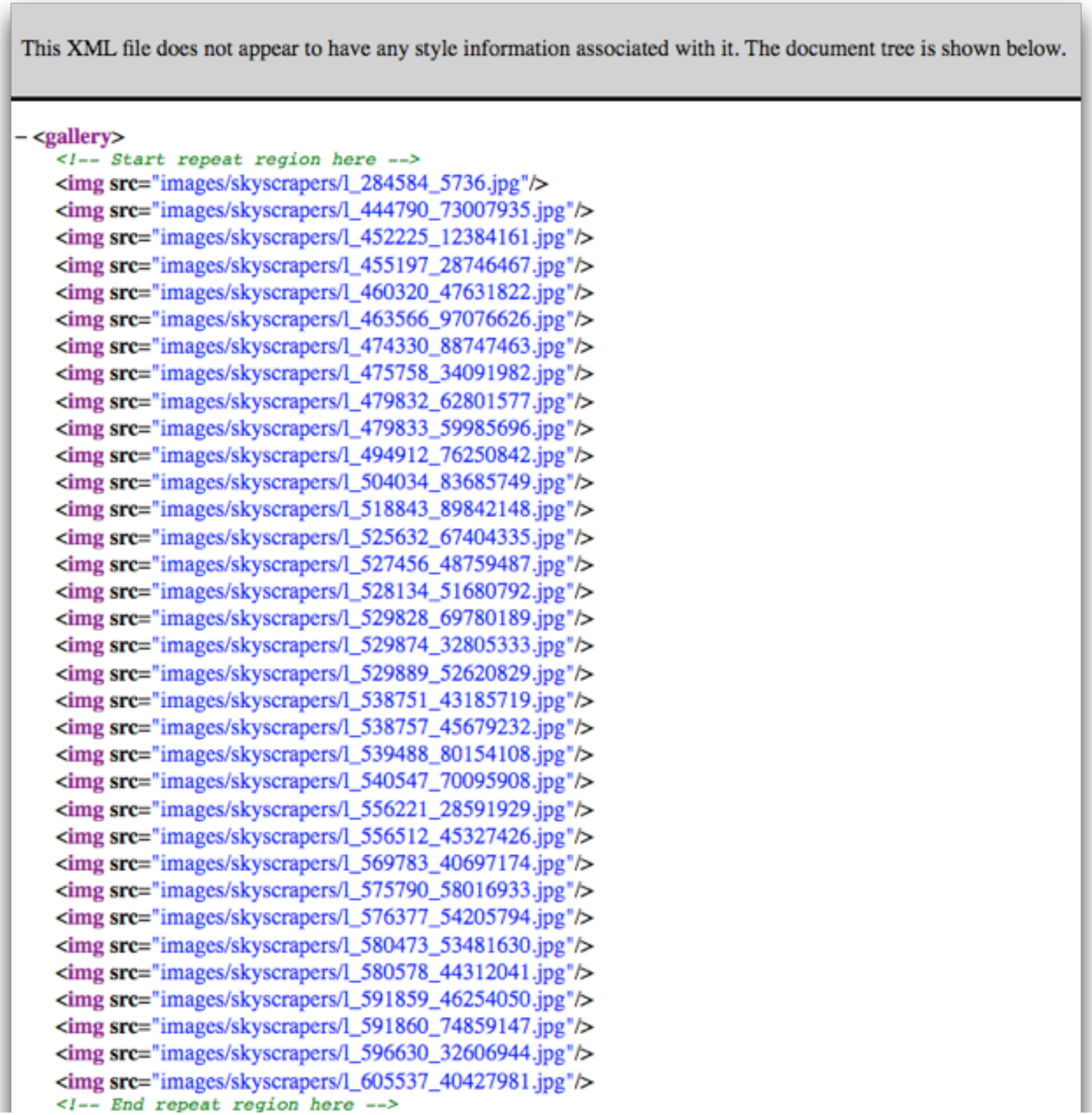
```
<?php echo($relativePathTo . $imageName); ?>
```

Your `<img ... />` line should now look like this:



```
<img src="<?php echo($relativePathTo . $imageName); ?>" />
```

You can now preview the file in your browser. As mentioned in the recordset example, Firefox is recommended. You should see something similar to this assuming you have defined the directory paths correctly:



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
– <gallery>
    <!-- Start repeat region here -->
    <img src="images/skyscrapers/l_284584_5736.jpg"/>
    <img src="images/skyscrapers/l_444790_73007935.jpg"/>
    <img src="images/skyscrapers/l_452225_12384161.jpg"/>
    <img src="images/skyscrapers/l_455197_28746467.jpg"/>
    <img src="images/skyscrapers/l_460320_47631822.jpg"/>
    <img src="images/skyscrapers/l_463566_97076626.jpg"/>
    <img src="images/skyscrapers/l_474330_88747463.jpg"/>
    <img src="images/skyscrapers/l_475758_34091982.jpg"/>
    <img src="images/skyscrapers/l_479832_62801577.jpg"/>
    <img src="images/skyscrapers/l_479833_59985696.jpg"/>
    <img src="images/skyscrapers/l_494912_76250842.jpg"/>
    <img src="images/skyscrapers/l_504034_83685749.jpg"/>
    <img src="images/skyscrapers/l_518843_89842148.jpg"/>
    <img src="images/skyscrapers/l_525632_67404335.jpg"/>
    <img src="images/skyscrapers/l_527456_48759487.jpg"/>
    <img src="images/skyscrapers/l_528134_51680792.jpg"/>
    <img src="images/skyscrapers/l_529828_69780189.jpg"/>
    <img src="images/skyscrapers/l_529874_32805333.jpg"/>
    <img src="images/skyscrapers/l_529889_52620829.jpg"/>
    <img src="images/skyscrapers/l_538751_43185719.jpg"/>
    <img src="images/skyscrapers/l_538757_45679232.jpg"/>
    <img src="images/skyscrapers/l_539488_80154108.jpg"/>
    <img src="images/skyscrapers/l_540547_70095908.jpg"/>
    <img src="images/skyscrapers/l_556221_28591929.jpg"/>
    <img src="images/skyscrapers/l_556512_45327426.jpg"/>
    <img src="images/skyscrapers/l_569783_40697174.jpg"/>
    <img src="images/skyscrapers/l_575790_58016933.jpg"/>
    <img src="images/skyscrapers/l_576377_54205794.jpg"/>
    <img src="images/skyscrapers/l_580473_53481630.jpg"/>
    <img src="images/skyscrapers/l_580578_44312041.jpg"/>
    <img src="images/skyscrapers/l_591859_46254050.jpg"/>
    <img src="images/skyscrapers/l_591860_74859147.jpg"/>
    <img src="images/skyscrapers/l_596630_32606944.jpg"/>
    <img src="images/skyscrapers/l_605537_40427981.jpg"/>
    <!-- End repeat region here -->
```

```
</gallery>
```

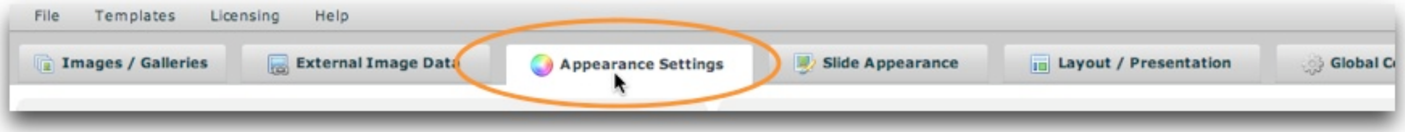
As intended, the list of all files in the directory has been outputted. If you add or remove files then reload the page, only the currently-present images will be listed.

If you instead see a PHP error about *'Output already started at line N...'*, then make sure that you have no whitespace or other extraneous code before the very first `<?php` code block. Additionally, make sure that there aren't any linebreaks or other whitespace characters between the first couple of `<?php ... ?>` blocks.

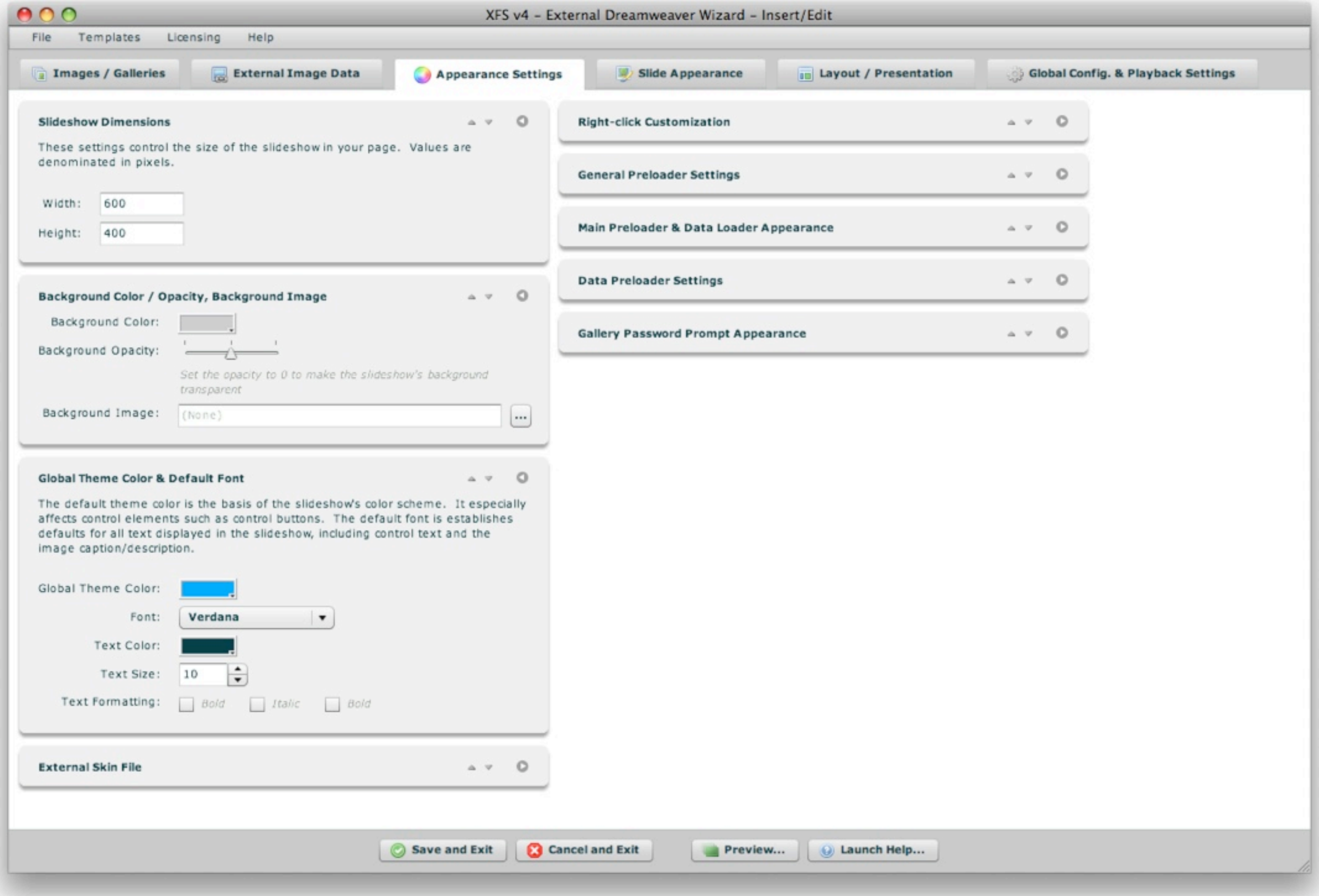
Once everything appears correctly, the dynamic data file is ready to be added in the wizard as a source of external image data.

Using the 'Appearance Settings' Section

The main Appearance Settings section is accessed by selecting the tab at the top of the wizard window:



The Appearance Settings section is where you will configure many settings which control the appearance of the overall slideshow. The section appears as follows:



Slideshow Dimensions

These settings control the size of the slideshow in your page. Values are denominated in pixels.

Width: 600

Height: 400

Background Color / Opacity, Background Image

Background Color:

Background Opacity:

Set the opacity to 0 to make the slideshow's background transparent

Background Image: (None)

Global Theme Color & Default Font

The default theme color is the basis of the slideshow's color scheme. It especially affects control elements such as control buttons. The default font is establishes defaults for all text displayed in the slideshow, including control text and the image caption/description.

Global Theme Color:

Font: Verdana

Text Color:

Text Size: 10

Text Formatting: □ *Bold* □ *Italic* □ *Bold*

External Skin File

Right-click Customization

General Preloader Settings

Main Preloader & Data Loader Appearance

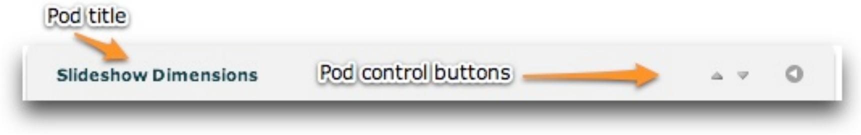
Data Preloader Settings

Gallery Password Prompt Appearance

Navigating the Section / Using the Pods

The Appearance Settings panel is made up of a number of collapsible pods. You can expand the pods which are most relevant to you, and re-order them as desired. The order and expanded/collapsed state of the pods is automatically saved, and the last view is restored each time you re-launch the wizard. This is designed to streamline the development process, instantly placing the most relevant and commonly used settings at your fingertips.

Each pod has several control buttons at the top right:



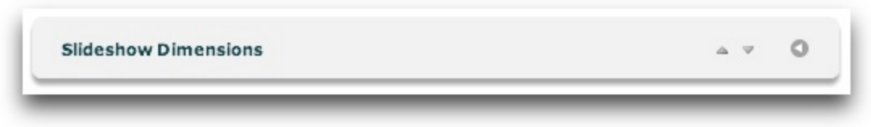
The first button allows you to move the pod up one space:

The second button allows you to move the pod down one space:

The third and final button allows you to expand or collapse the pod:

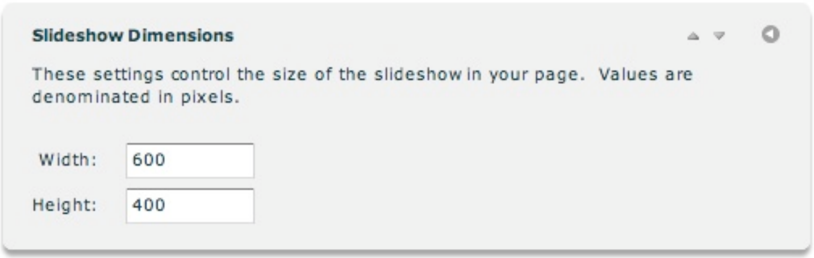
Tip: You can also expand or collapse a pod by clicking on its title.

When collapsed, a pod will appear like this:



The following sections of this topic detail each pod and the options each controls.

The Slideshow Dimensions Pod



Slideshow Dimensions

These settings control the size of the slideshow in your page. Values are denominated in pixels.

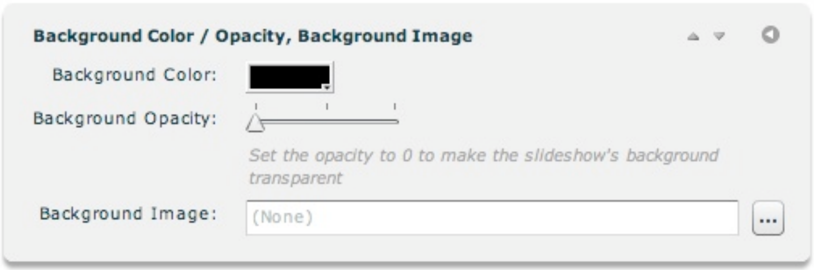
Width: 600

Height: 400

This is where you will specify the dimensions of the slideshow. This is the overall size of the slideshow element in your page. Both values are denominated in pixels. You can also use a percentage value (usually 100%), when creating a full-browser-window slideshow for example.

You can also modify the slideshow dimensions in the Property Inspector after you have inserted a slideshow.

The Background Color / Opacity, Background Image Pod



Background Color / Opacity, Background Image

Background Color:

Background Opacity:

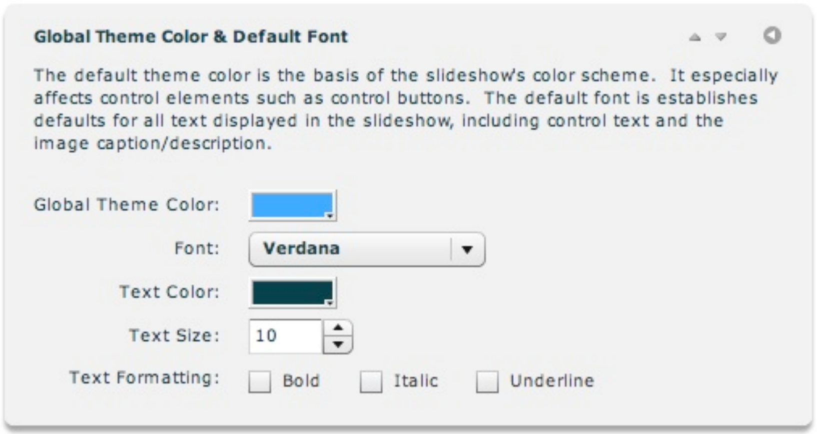
Set the opacity to 0 to make the slideshow's background transparent

Background Image: (None)

This pod contains the background appearance options for the slideshow. Option overview:

Property	Description
Background Color	The color of the background behind the slideshow. Only applicable if Background Opacity is greater than 0.
Background Opacity	Specifies the opacity (and transparency) of the background color. If you set the value to 0% opacity / 100% transparency, then the slideshow will be transparent. In this case, any content in the HTML page behind the slideshow will show through.
Background Image	Optionally specifies the image which should show as part of the slideshow background. Use the browse button on the right ('...') to select a JPEG, PNG, SWF, or GIF file. The URL must be relative to the HTML page holding the slideshow; using the browse button will automatically create this URL properly. A background image can be used much like a loading image, if you want a specific image to be visible while loading occurs.

The Global Theme Color and Default Font Pod



Global Theme Color & Default Font

The default theme color is the basis of the slideshow's color scheme. It especially affects control elements such as control buttons. The default font is establishes defaults for all text displayed in the slideshow, including control text and the image caption/description.

Global Theme Color:

Font: Verdana

Text Color:

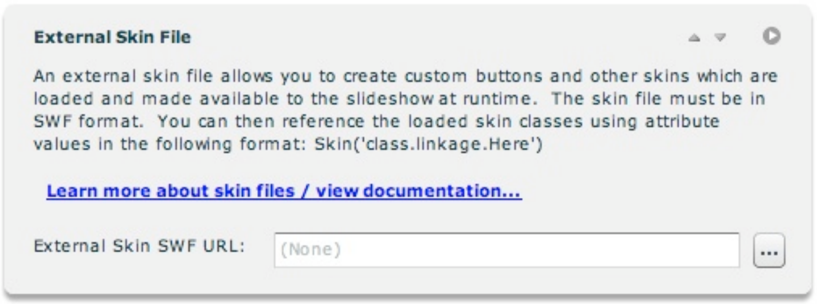
Text Size: 10

Text Formatting: □ Bold □ Italic □ Underline

The default theme color is the basis of the slideshow's color scheme. It especially affects control elements, such as control buttons' icon color. The default font establishes defaults for all text displayed in the slideshow, any control text and the image description/caption. The text size is denominated in pixels. The theme color and font settings can be overridden on any specific layout element using the Properties panel.

XML Style Setting Equivalents (for Advanced Users or Developers)

The External Skin File Pod



External Skin File

An external skin file allows you to create custom buttons and other skins which are loaded and made available to the slideshow at runtime. The skin file must be in SWF format. You can then reference the loaded skin classes using attribute values in the following format: Skin('class.linkage.Here')

Learn more about skin files / view documentation...

External Skin SWF URL: (None)

This pod allows you to specify a custom SWF skin file which contains skin assets for the slideshow. These skin files can be used, for example, to customize buttons and other skinnable assets. Icons can be embedded as well. All embedded skin and icon assets which have an export linkage class will then be accessible in the slideshow by using the following format in a Skin or Icon field:

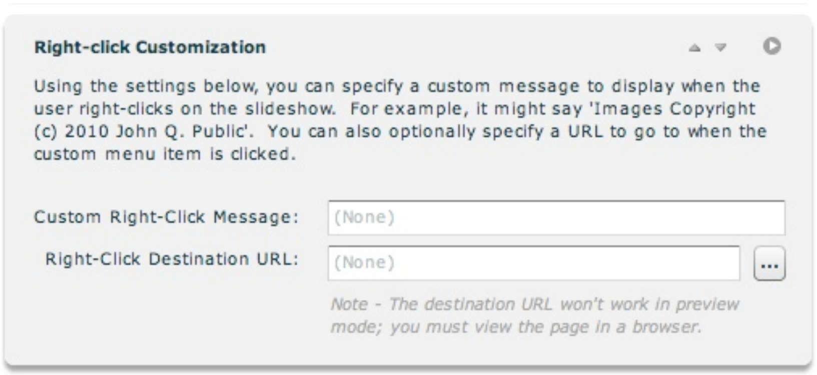
```
Skin('class.linkage.Here')
```

Where `class.linkage.Here` is the class name you used when setting the linkage settings for the clip.

To learn more about creating custom skins, see the Custom Skin Files / Custom Buttons topic.

The URL must be relative to your local file, and should be hosted on the same server (accessible by a relative URL) to ensure that you don't experience Flash Player cross-domain loading restrictions.

The Right-Click Customization Pod



Right-click Customization

Using the settings below, you can specify a custom message to display when the user right-clicks on the slideshow. For example, it might say 'Images Copyright (c) 2010 John Q. Public'. You can also optionally specify a URL to go to when the custom menu item is clicked.

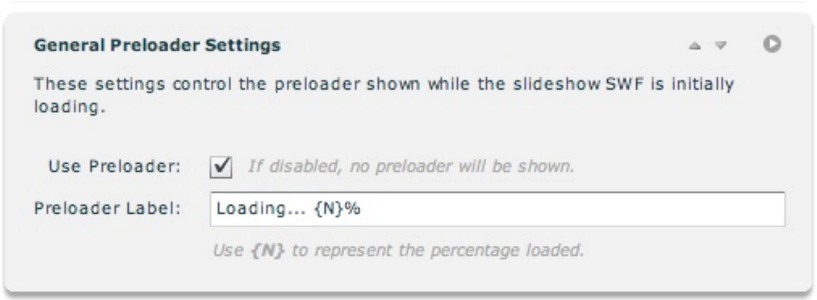
Custom Right-Click Message: (None)

Right-Click Destination URL: (None)

Note - The destination URL won't work in preview mode; you must view the page in a browser.

By default, right-clicking on the slideshow only shows 'Settings...' (disabled) and 'About Adobe Flash Player...'. If you prefer, you can customize this menu by adding a custom right click message in the 'Custom Right-Click Message' field. This should be a short string value, such as 'Images Copyright (c) 2010 John Q. Public'. If you specify only a message value, the message will appear disabled in the context menu. If you specify a 'Right-Click Destination URL' as well (e.g. `http://www.mysite.com/somepage.html`), the message will be enabled and clicking on it will redirect to that URL. Use the browse button on the right ('...') to select a file directly.

The General Preloader Settings Pod



General Preloader Settings

These settings control the preloader shown while the slideshow SWF is initially loading.

Use Preloader: ☑ *If disabled, no preloader will be shown.*

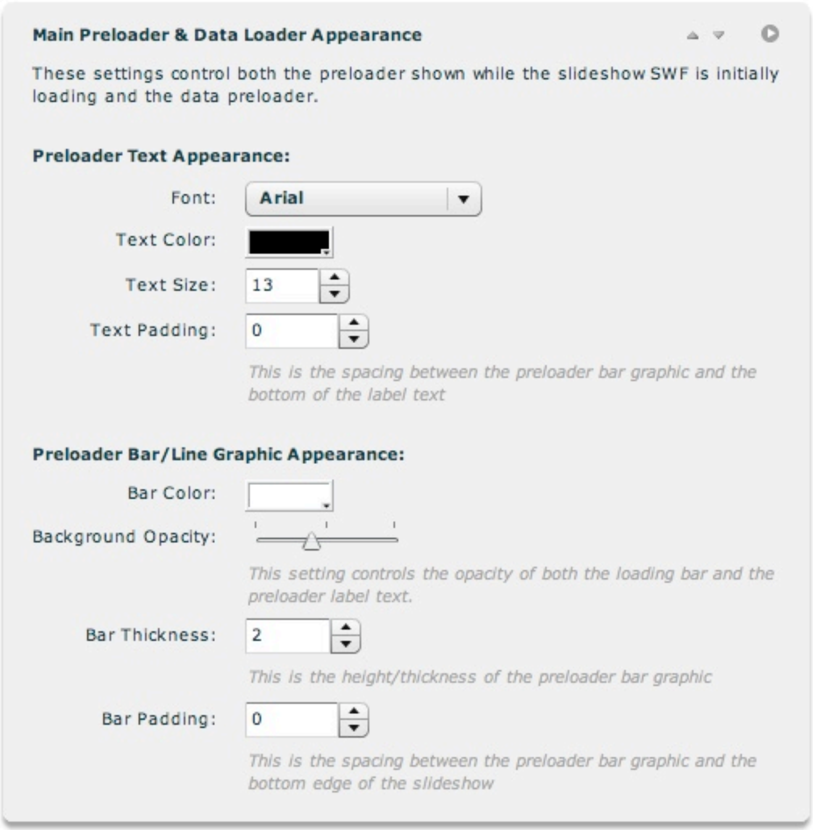
Preloader Label: Loading... {N}%

*Use **{N}** to represent the percentage loaded.*

The settings in this pod control the preloader which is displayed while the slideshow SWF is initially loading.

Property	Description
Use Preloader	Specifies if a preloader should be shown at all. If disabled, no preloader will be shown. You can disable the default preloader and specify a background image if you want a custom preloader.
Preloader Label	If the preloader has been enabled, this controls the label on the preloader. Wherever `{N}` is included in the label, it will be replaced with the currently loaded percentage. For example, the default label value would show *'Loading... 50%'* halfway through loading.

The Main Preloader and Data Loader Appearance Pod



Main Preloader & Data Loader Appearance

These settings control both the preloader shown while the slideshow SWF is initially loading and the data preloader.

Preloader Text Appearance:

Font: Arial

Text Color:

Text Size: 13

Text Padding: 0

This is the spacing between the preloader bar graphic and the bottom of the label text

Preloader Bar/Line Graphic Appearance:

Bar Color:

Background Opacity:

This setting controls the opacity of both the loading bar and the preloader label text.

Bar Thickness: 2

This is the height/thickness of the preloader bar graphic

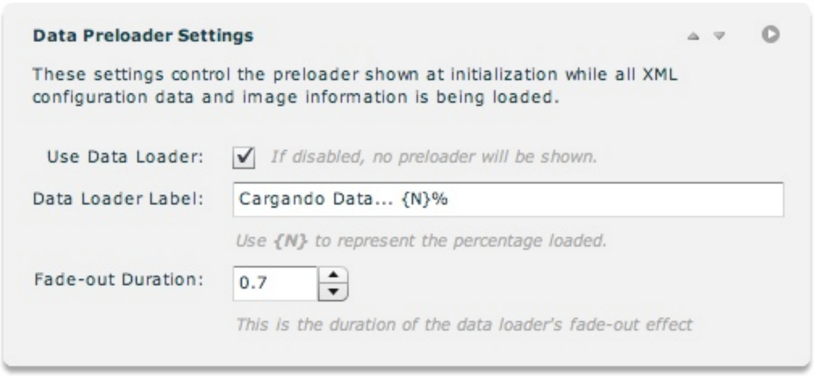
Bar Padding: 0

This is the spacing between the preloader bar graphic and the bottom edge of the slideshow

The settings in this pod control the appearance of all preloaders -- both the one shown while the slideshow SWF is initially loading and the data preloader.

Property	Description
Font / Text Color / Text Size	These settings control the appearance of the preloader text. Text size is denominated in pixels.
Text Padding	Specifies the spacing (in pixels) between the preloader bar graphic and the bottom of the label text.
Bar Color	The color of the preloader bar which grows in size.
Bar Opacity	The opacity (and transparency) of both the preloader bar **and** the preloader label text. A lower value makes the preloader less obtrusive, but also less noticeable.
Bar Thickness	Specifies the height/thickness of the preloader bar in pixels
Bar Padding	Specifies the spacing, in pixels, between the preloader bar graphic and the bottom edge of the slideshow. The default value (0) positions the bar directly agains the bottom of the slideshow.

The Data Preloader Settings Pod



Data Preloader Settings

These settings control the preloader shown at initialization while all XML configuration data and image information is being loaded.

Use Data Loader: ☑ *If disabled, no preloader will be shown.*

Data Loader Label: Cargando Data... {N}%

Use {N} to represent the percentage loaded.

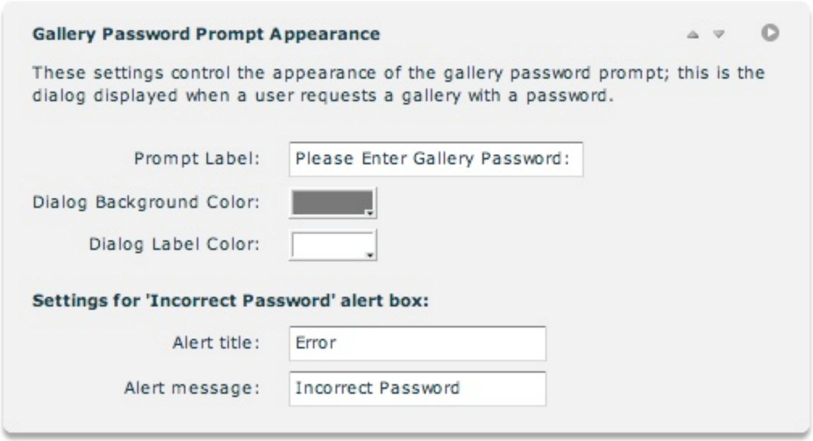
Fade-out Duration: 0.7

This is the duration of the data loader's fade-out effect

The settings in this pod control the preloader shown at initialization while all XML configuration data and image information (any external image data) is being loaded.

Property	Description
Use Data Loader	Specifies if a data preloader should be shown at all. If disabled, no data preloader will be shown. You can disable the default data preloader and specify a background image if you want a custom data preloader.
Data Loader Label	If the data preloader has been enabled, this controls the label on the data preloader. Wherever `{N}` is included in the label, it will be replaced with the currently loaded percentage. For example, the default label value would show *'Loading Data... 50%'* halfway through loading.
Fade-out Duration	When data loading completes, the data preloader quickly fades out prior to the slideshow initializing. You can customize the length of this fade-out effect, or essentially disable it by setting it to 0 or 0.1. The value is denominated in seconds.

The Gallery Password Prompt Apperance Pod



Gallery Password Prompt Appearance

These settings control the appearance of the gallery password prompt; this is the dialog displayed when a user requests a gallery with a password.

Prompt Label: Please Enter Gallery Password:

Dialog Background Color:

Dialog Label Color:

Settings for 'Incorrect Password' alert box:

Alert title: Error

Alert message: Incorrect Password

The settings in this pod control the appearance of the gallery password prompt. This is the dialog displayed when a user request a gallery which has been password-protected.

Property	Description
Prompt Label	This is the label displayed in the password prompt dialog, which asks for the user to enter the gallery password. Generally, you would only want to modify the value if you need to translate it to another language.
Dialog Background Color	Specifies the background color of the password prompt dialog.
Dialog Label Color	Specifies the color of the text in the password prompt dialog.
Alert Title	Specifies the title of the alert box displayed if a user incorrectly enters the gallery password. Generally, you would only want to modify this value if you need to translate it to another language.
Alert Message	Specifies the message in the alert box displayed if a user incorrectly enters the gallery password. Generally, you would only want to modify this value if you need to translate it to another language.

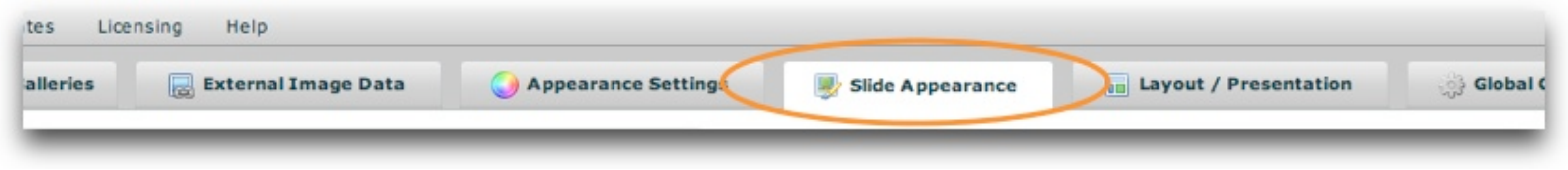
XML Style Setting Equivalents (for Advanced Users or Developers)

Developer Note

Most of the settings in this section are initialization settings, passed through the slideshow HTML. For detailed reference on all initialization parameters, see the `com.dwuser.ss4.core.FlashSlideshowConfigurationParameters` class in the API Reference.

Using the 'Slide Appearance' Section

The main Slide Appearance section is accessed by selecting the tab at the top of the wizard window:



The Slide Appearance section is where you will configure many settings which control the appearance of each of the photos / slides in the slideshow. The section appears as follows:

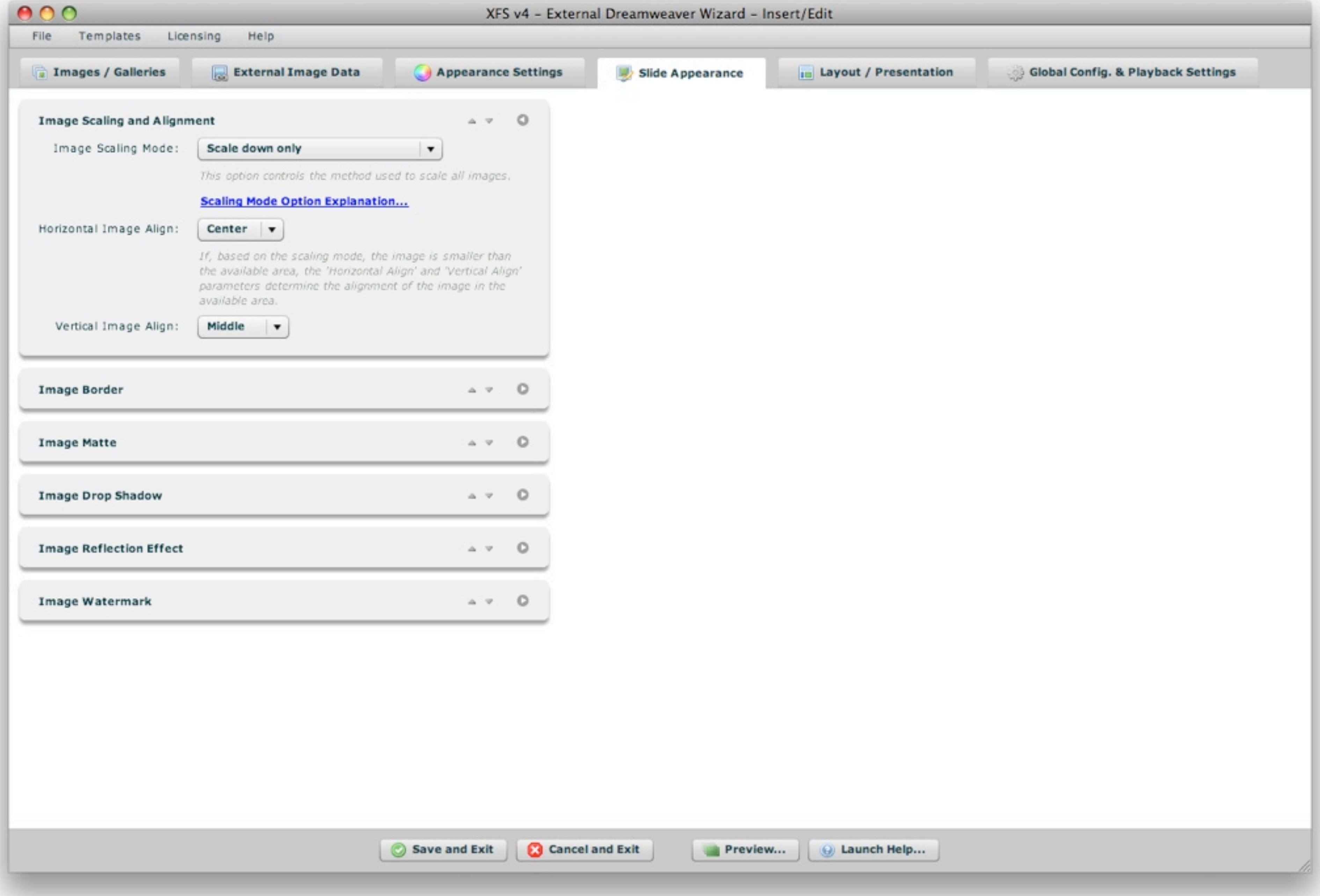


Image Scaling and Alignment

Image Scaling Mode: Scale down only

This option controls the method used to scale all images.

Scaling Mode Option Explanation...

Horizontal Image Align: Center

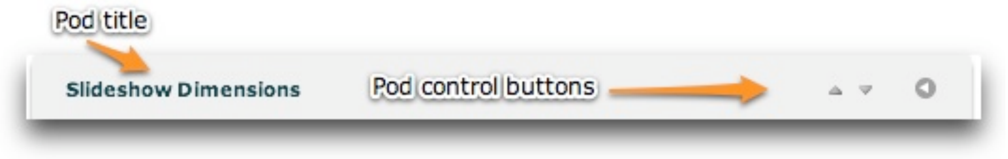
If, based on the scaling mode, the image is smaller than the available area, the 'Horizontal Align' and 'Vertical Align' parameters determine the alignment of the image in the available area.

Vertical Image Align: Middle

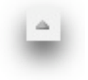
Navigating the Section / Using the Pods

The Slide Appearance panel is made up of a number of collapsible pods. You can expand the pods which are most relevant to you, and re-order them as desired. The order and expanded/collapsed state of the pods is automatically saved, and the last view is restored each time you re-launch the wizard. This is designed to streamline the development process, instantly placing the most relevant and commonly used settings at your fingertips.

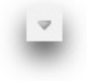
Each pod has several control buttons at the top right:



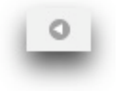
The first button allows you to move the pod up one space:



The second button allows you to move the pod down one space:

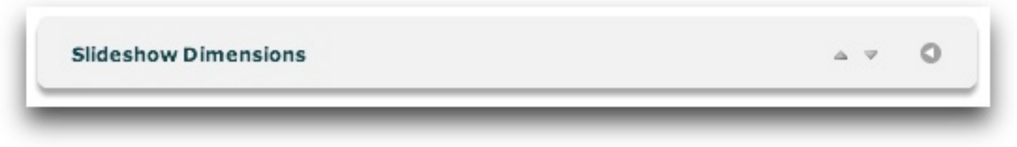


The third and final button allows you to expand or collapse the pod:



Tip: You can also expand or collapse a pod by clicking on its title.

When collapsed, a pod will appear like this:



The following sections of this topic detail each pod and the options each controls.

The Image Scaling and Alignment Pod

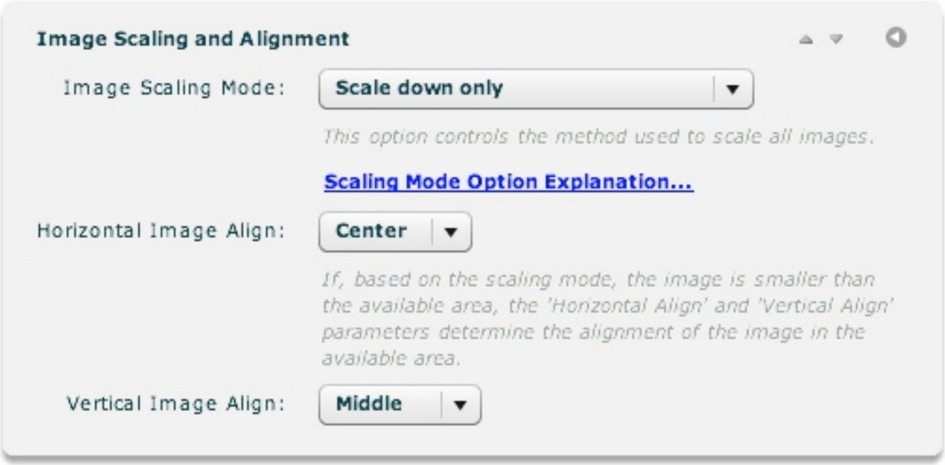


Image Scaling and Alignment

Image Scaling Mode: Scale down only

This option controls the method used to scale all images.

Scaling Mode Option Explanation...

Horizontal Image Align: Center

If, based on the scaling mode, the image is smaller than the available area, the 'Horizontal Align' and 'Vertical Align' parameters determine the alignment of the image in the available area.

Vertical Image Align: Middle

The settings in this pod control how the photos on each slide are scaled and aligned. If, based on the scaling mode, the image is smaller than the available area, the Horizontal Image Align and Vertical Image Align settings determine the alignment of the image within the available area.

Property	Description
Image Scaling Mode	Specifies how the slide images are scaled. The available scaling modes are: ▪ **No Scaling / Show as much as fits** - (`showAvailable` XML attribute value) - No scaling; images are displayed at their original size, and cropping occurs if images are larger than the available area. ▪ **Scale Down Only** - (`scaleDown` XML attribute value) - Images are scaled down to fit within the available area while maintaining their aspect ratio. If the images are smaller than the available area, their original size is maintained. ▪ **Scale Both Up and Down** - (`scaleUpAndDown` XML attribute value) - Images are scaled both up and down as appropriate, so that they are as large as possible while being fully visible in the available area. ▪ **Always Scale to Fully Fill Area** - (`fillArea` XML attribute value) - Images are scaled both up and down as appropriate, so that they always fully fill the available area. If the aspect ratio of the available area differs from that of an image, one of the image's dimensions is cropped.
Horizontal Image Align	If, based on the scaling mode, the image width is smaller than the available width, this determines the horizontal alignment of the image within the available area.
Vertical	If, based on the scaling mode, the image height is smaller than the available height, this determines the vertical alignment of the image within

Image Align	the available area.

XML Style Setting Equivalents (for Advanced Users or Developers)

The Image Border Pod

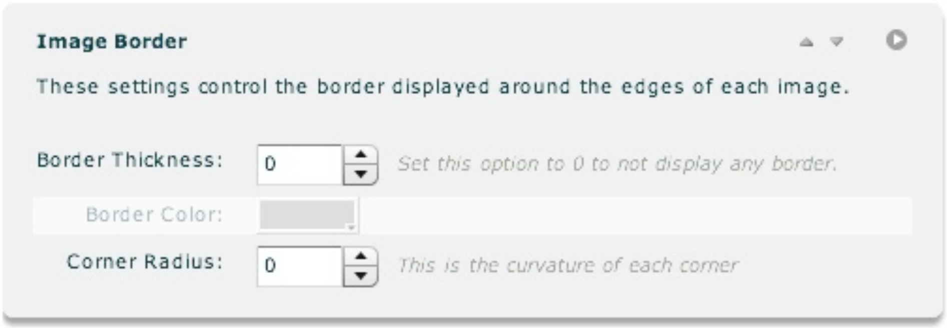


Image Border

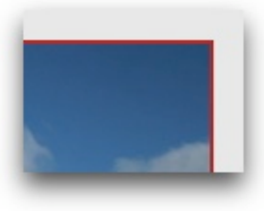
These settings control the border displayed around the edges of each image.

Border Thickness: 0 — Set this option to 0 to not display any border.

Border Color:

Corner Radius: 0 — This is the curvature of each corner

The settings in this pod control the border displayed around the edges of each image. Here is an example of an image with a 2px red border:



Once you set the Border Thickness to a value of greater than 0 (which enables the border), the Border Color option becomes enabled:

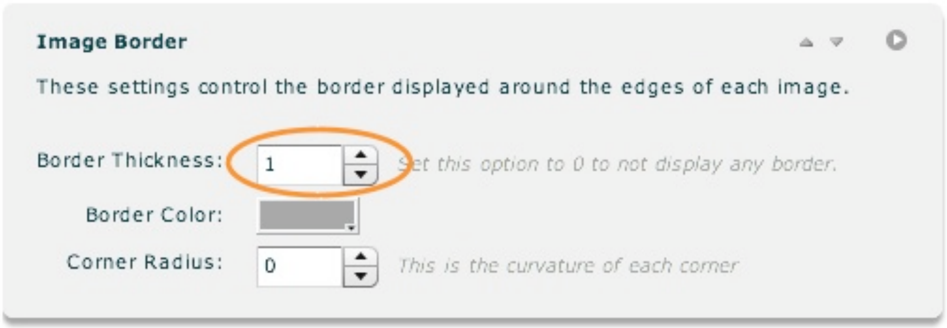


Image Border

These settings control the border displayed around the edges of each image.

Border Thickness: 1 — Set this option to 0 to not display any border.

Border Color:

Corner Radius: 0 — This is the curvature of each corner

Option details:

Property	Description
Border Thickness	The width, in pixels, of the border around the image on each slide. Using a value of 0 will disable the border.
Border Color	If a border has been configured to show (Border Thickness > 0), specifies the color of that border.
Corner Radius	Specifies the radius of curvature (in pixels) of the border if enabled.

XML Style Setting Equivalents (for Advanced Users or Developers)

The Image Matte Pod

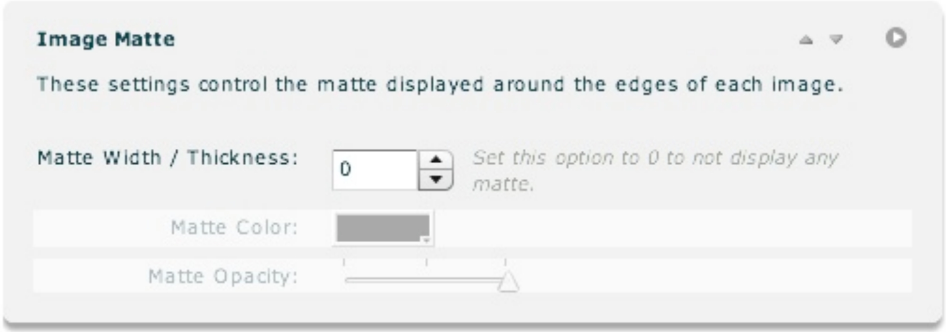


Image Matte

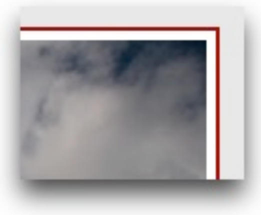
These settings control the matte displayed around the edges of each image.

Matte Width / Thickness: 0 *Set this option to 0 to not display any matte.*

Matte Color:

Matte Opacity:

The settings in this pod control the matte displayed around the edges of each image. Here is an example of an image with a 5px white matte and 2px red border:



Mattes are essentially an "inner border" if used in conjunction with a regual border (as defined in the Image Border pod). If you enable a matte by setting the Matte Width / Thickness to a value of 1 or greater, the other two option will become enabled:

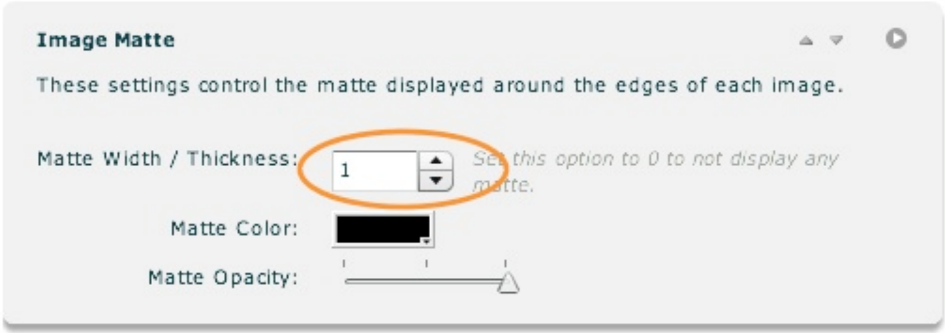


Image Matte

These settings control the matte displayed around the edges of each image.

Matte Width / Thickness: 1 *Set this option to 0 to not display any matte.*

Matte Color:

Matte Opacity:

Option details:

Property	Description
Matte Width / Thickness	The width, in pixels, of the matte around the image on each slide. Using a value of 0 will disable the matte.
Matte Color	If a matte has been configured to show (Matte Width / Thickness > 0), specifies the color of that matte.
Matte Opacity	Specifies the opacity/transparency of the matte if enabled.

XML Style Setting Equivalents (for Advanced Users or Developers)

The Image Drop Shadow Pod

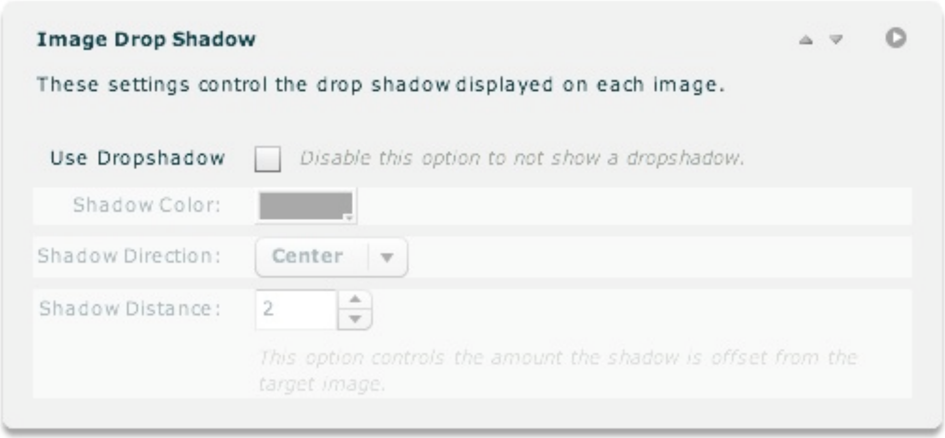


Image Drop Shadow

These settings control the drop shadow displayed on each image.

Use Dropshadow □ *Disable this option to not show a dropshadow.*

Shadow Color:

Shadow Direction: Center

Shadow Distance: 2

This option controls the amount the shadow is offset from the target image.

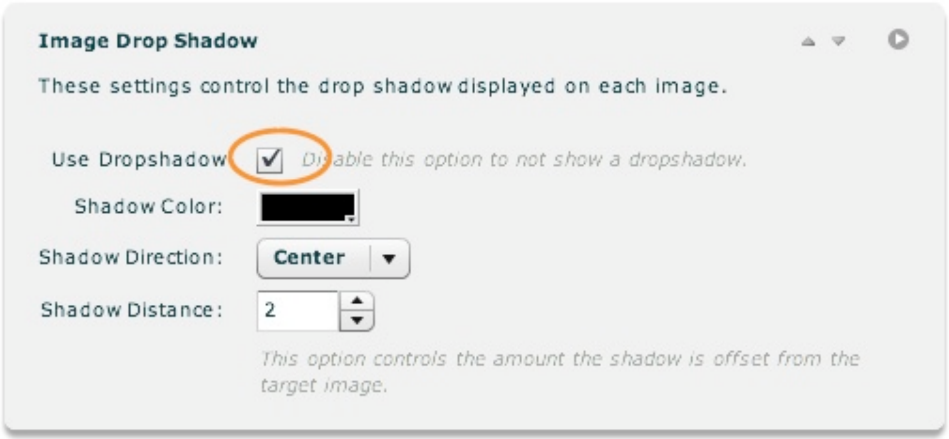


Image Drop Shadow

These settings control the drop shadow displayed on each image.

Use Dropshadow ☑ *Disable this option to not show a dropshadow.*

Shadow Color: ■

Shadow Direction: Center

Shadow Distance: 2

This option controls the amount the shadow is offset from the target image.

The settings in this pod allow you to add a drop-shadow to each slide. Once you enable the 'Use Dropshadow' option, all of the other options become enabled. Here is an example of a slide with a drop shadow:

Option details:

Property	Description
Use Dropshadow	Specifies if a dropshadow should be shown on each slide
Shadow Color	If a drop-shadow has been enabled, specifies the color of the shadow
Shadow Direction	If a drop-shadow has been enabled, specifies the direction of the shadow. This value can effectively be reversed by using a negative Shadow Distance. Supported values are: - **Left** - (`left` XML attribute value) - Shadow appears on the left, as if the light is from the right. - **Center** - (`center` XML attribute value) - Shadow appears below the object, as if the light is from above. - **Right** - (`right` XML attribute value) - Shadow appears on the right, as if the light is from the left.
Shadow Distance	If a drop-shadow has been enabled, specifies the number of pixels the shadow is offset from the slide. Negative values can be used to reverse the shadow direction.

XML Style Setting Equivalents (for Advanced Users or Developers)

The Image Reflection Effect Pod

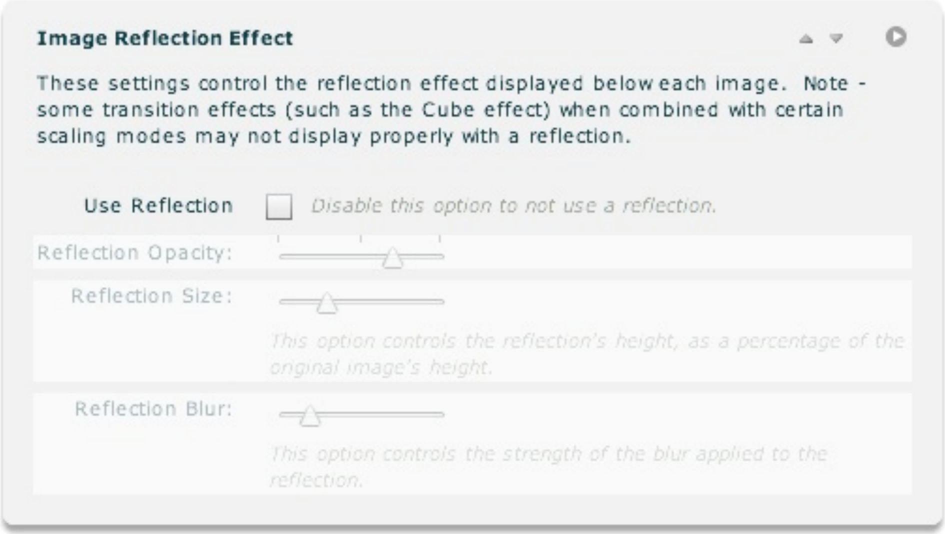


Image Reflection Effect

These settings control the reflection effect displayed below each image. Note - some transition effects (such as the Cube effect) when combined with certain scaling modes may not display properly with a reflection.

Use Reflection □ *Disable this option to not use a reflection.*

Reflection Opacity:

Reflection Size:

This option controls the reflection's height, as a percentage of the original image's height.

Reflection Blur:

This option controls the strength of the blur applied to the reflection.

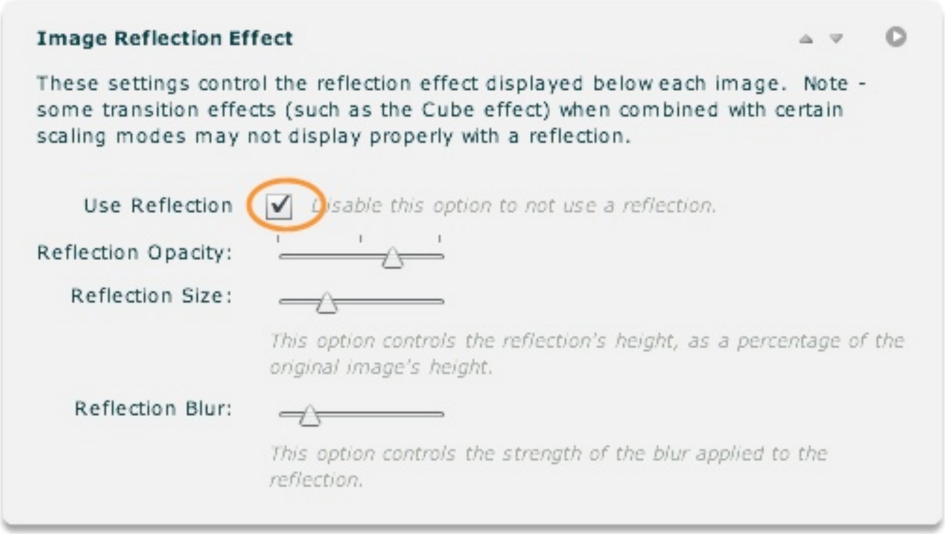


Image Reflection Effect

These settings control the reflection effect displayed below each image. Note - some transition effects (such as the Cube effect) when combined with certain scaling modes may not display properly with a reflection.

Use Reflection ☑ *Disable this option to not use a reflection.*

Reflection Opacity:

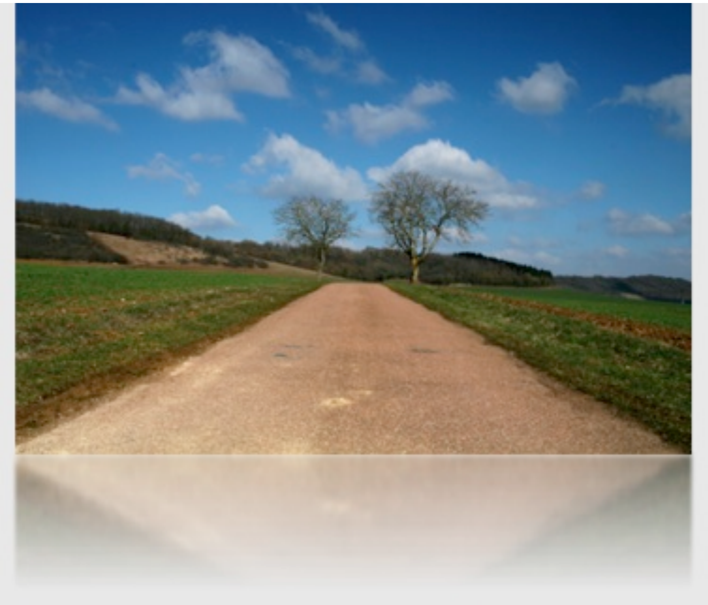
Reflection Size:

This option controls the reflection's height, as a percentage of the original image's height.

Reflection Blur:

This option controls the strength of the blur applied to the reflection.

The settings in this pod allow you to add a reflection effect to each slide. Once you enable the 'Use Reflection' option, all of the other options become enabled. Here is an example of a slide with a reflection effect applied:



Option details:

Property	Description
Use Reflection	Specifies if a reflection should be shown on each slide.
Reflection Opacity	If a reflection is enabled, specifies the opacity/transparency of the reflection. A lower opacity will result in a more subtle effect.
Reflection Size	If a reflection is enabled, controls the height of the reflection as a percentage of the original image's height.
Reflection Blur	If a reflection is enabled, controls the strength of the blur applied to the reflection. Set to 0 to disable blurring.

XML Style Setting Equivalents (for Advanced Users or Developers)

The Image Watermark Pod

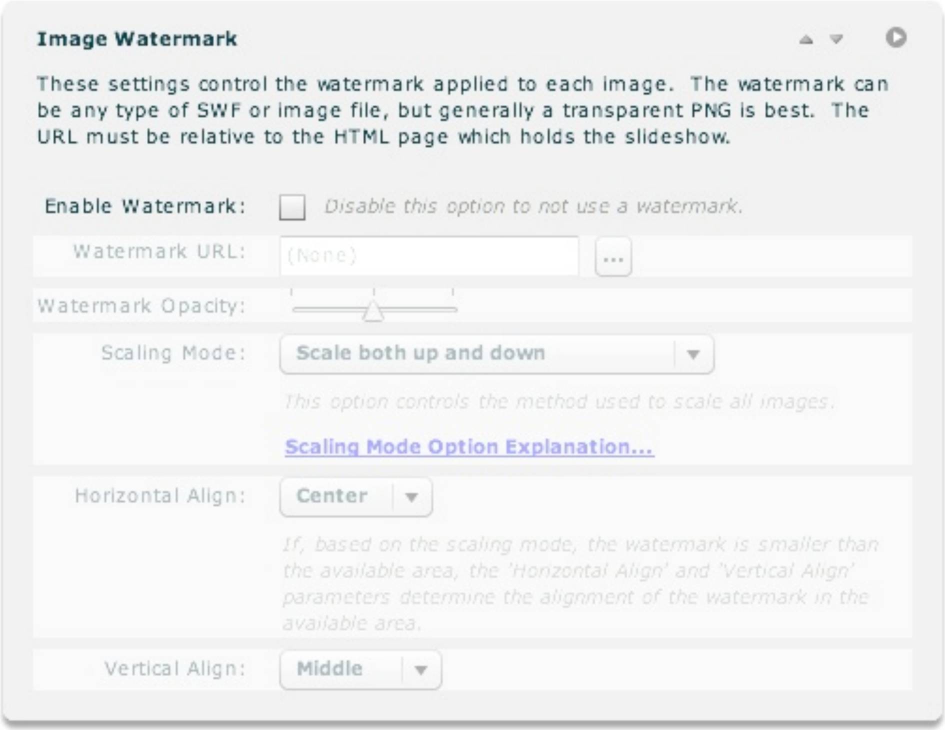


Image Watermark

These settings control the watermark applied to each image. The watermark can be any type of SWF or image file, but generally a transparent PNG is best. The URL must be relative to the HTML page which holds the slideshow.

Enable Watermark: □ *Disable this option to not use a watermark.*

Watermark URL: (None) ...

Watermark Opacity:

Scaling Mode: Scale both up and down ▾

This option controls the method used to scale all images.

Scaling Mode Option Explanation...

Horizontal Align: Center ▾

If, based on the scaling mode, the watermark is smaller than the available area, the 'Horizontal Align' and 'Vertical Align' parameters determine the alignment of the watermark in the available area.

Vertical Align: Middle ▾

The settings in this pod allow you to add a watermark to each slide. This is another image (JPEG, PNG, SWF, or GIF) which overlays every main slide, displaying ownership information etc. Once you enable the 'Enable Watermark' option, all of the other options become enabled:

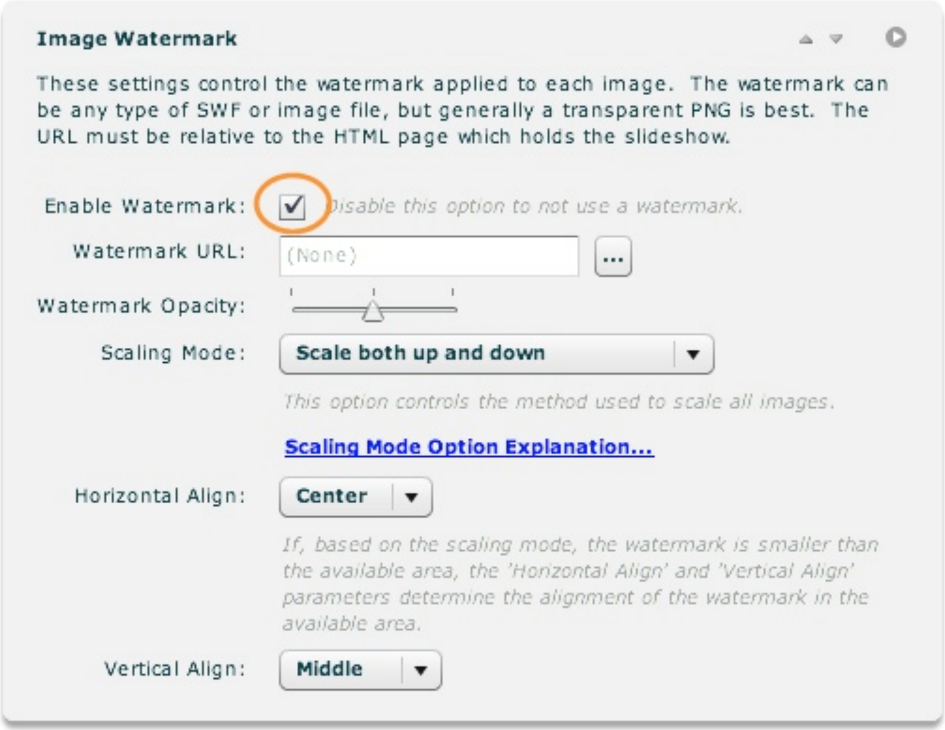


Image Watermark

These settings control the watermark applied to each image. The watermark can be any type of SWF or image file, but generally a transparent PNG is best. The URL must be relative to the HTML page which holds the slideshow.

Enable Watermark: ☑ Disable this option to not use a watermark.

Watermark URL: (None)

Watermark Opacity:

Scaling Mode: Scale both up and down

This option controls the method used to scale all images.

Scaling Mode Option Explanation...

Horizontal Align: Center

If, based on the scaling mode, the watermark is smaller than the available area, the 'Horizontal Align' and 'Vertical Align' parameters determine the alignment of the watermark in the available area.

Vertical Align: Middle

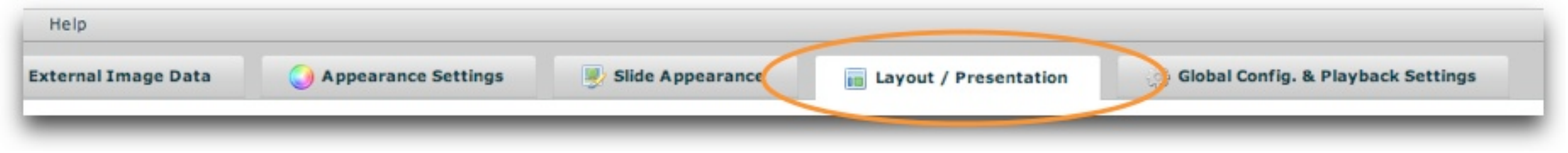
Option details:

Property	Description
Enable Watermark	Specifies if a watermark should be used. If you enable this option, you must then specify a Watermark URL.
Watermark URL	If a watermark is enabled, specifies the URL of the watermark overlay image. This file can be a JPEG, PNG, SWF, or GIF. The path must be relative to the HTML page holding the slideshow; such a path will be generated automatically if you use the browse button ('...') to the right of the field.
Watermark Opacity	If a watermark is enabled, specifies the opacity/transparency of the overlaid image. A lower opacity will result in a more subtle watermark.
Scaling Mode	If a watermark is enabled, specifies the scaling mode used for the overlay image. To learn more about the available scaling modes, see the "Image Scaling Mode" parameter in the **The Image Scaling and Alignment Pod** section above.
Horizontal Align	Only applicable if a watermark is enabled. If, based on the watermark scaling mode, the watermark is smaller than the available area, this setting controls the horizontal alignment of the overlaid image within the available area.
Vertical Align	Only applicable if a watermark is enabled. If, based on the watermark scaling mode, the watermark is smaller than the available area, this setting controls the vertical alignment of the overlaid image within the available area.

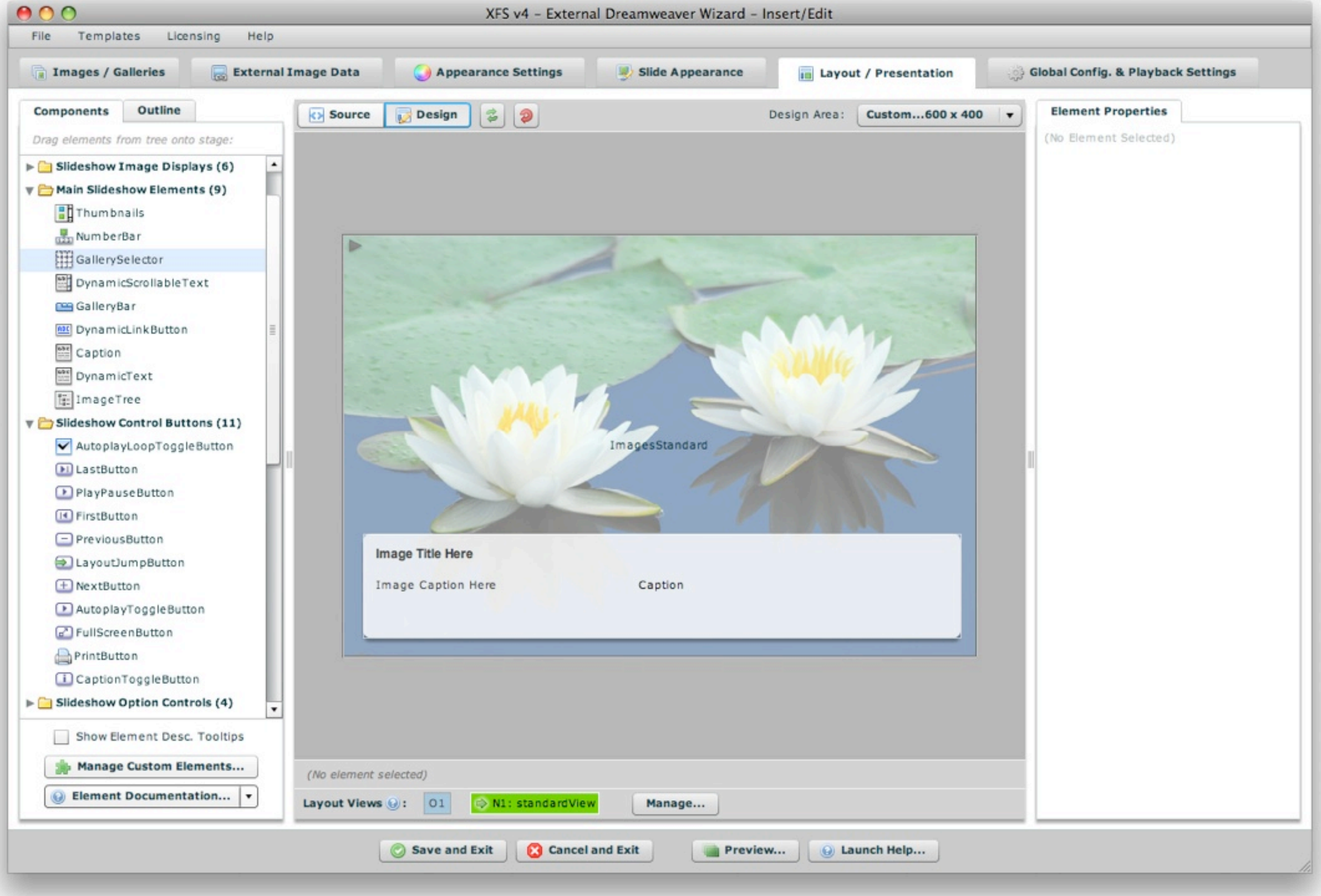
XML Style Setting Equivalents (for Advanced Users or Developers)

Using the 'Layout / Presentation' Section

The main Layout / Presentation section is accessed by selecting the tab at the top of the wizard window:



The Layout / Presentation section is where you will create and/or edit the slideshow's layout and the general user interface for the slideshow. The section appears as follows:



Drag elements from tree onto stage:

Design Area: Custom...600 x 400

(No Element Selected)

☐ Show Element Desc. Tooltips

(No element selected)

Layout Views: 01 N1: standardView

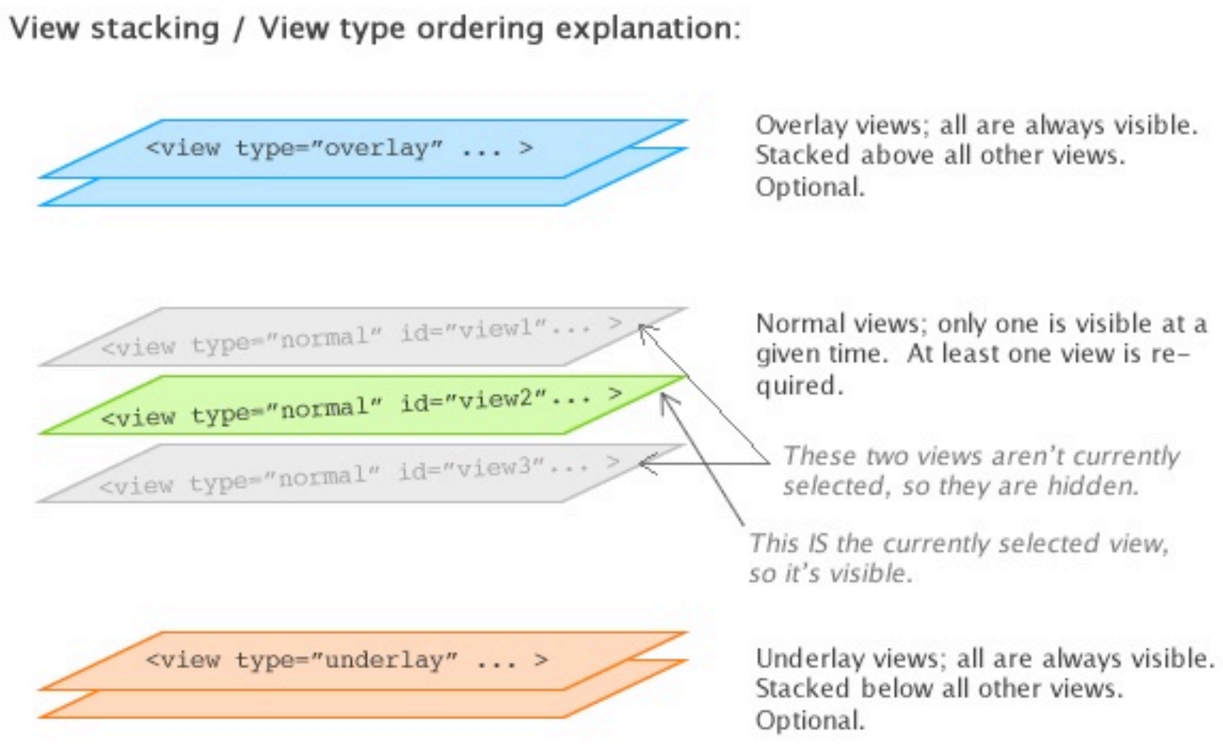
The subtopics in this section detail the process of user interface creation using the wizard.   Make sure to read all of the topics for a thorough understanding of building, maintaining, and customizing slideshow layouts / user interfaces.

Understanding and Managing Layout Views

Slideshow layouts are made up of one or more "screens" which are referred to as **views**. There are three different view types:

- **Overlay Views** - Optional; any number can be used. Overlay views are always visible. They are outside of the normal changeable views, resting at the very top of the visual stack, always overlaying all the other views. They provide a way to float elements on top of all other layout elements. For example, you will usually place the SingleLoader and GroupLoader elements (which display [pre]loading progress) in an overlay view, so that no matter what the current layout state is, the user will be notified of the loading status.
- **Normal Views** - At least one is required. These views are positioned between the Overlay and Underlay views, and only one is visible at a given time. They generally hold the main content. Because only one is visible at a given time (and you can configure the transition between each), these views allow you to create multiple "screens" within the slideshow, each with a different appearance and a unique set of constituent elements. When transitioning, you can use simple cut effects or advanced visual effects like views 'popping' in and out, or a cube rotation. See Using the 'Global Configuration and Playback Settings' Section for more information about transitions.
- **Underlay Views** - Optional; any number can be used. Underlay views are always visible. They are outside of the normal changeable views, resting at the very bottom of the visual stack, always below all the other views. You can have one or more underlay views. They provide a way to display constant background elements which should always be visible, regardless of which normal view is currently visible.

The following diagram provides a visual representation of how views function:



View stacking / View type ordering explanation:

Overlay views; all are always visible. Stacked above all other views. Optional.

Normal views; only one is visible at a given time. At least one view is required.

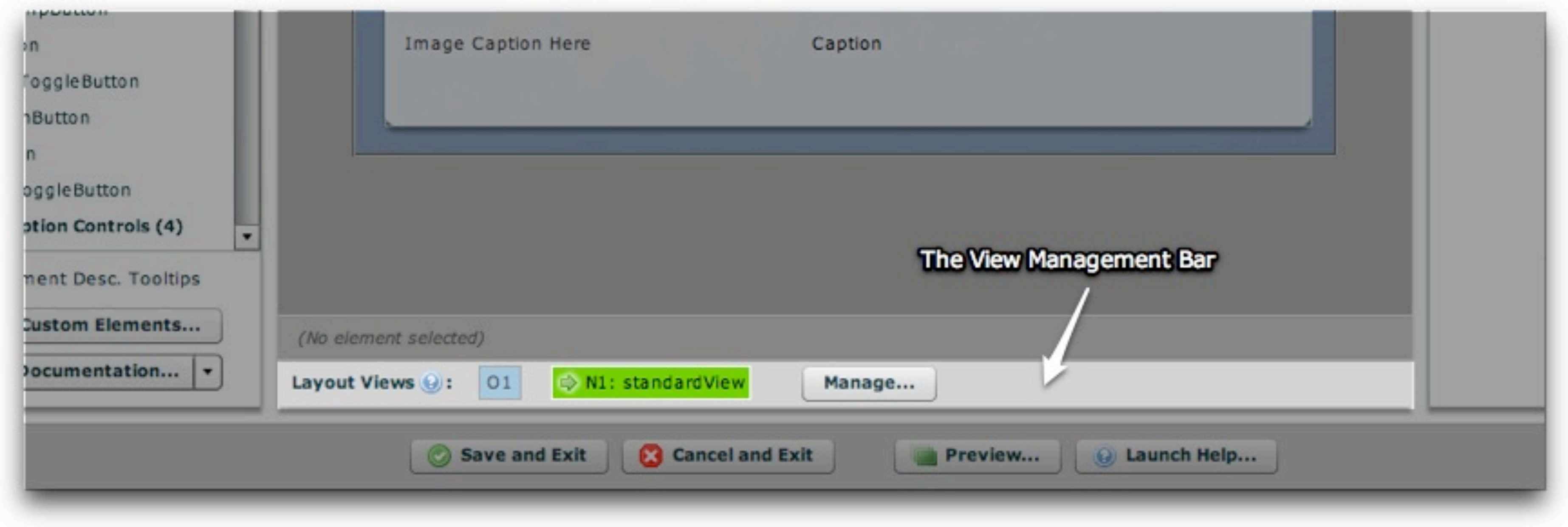
These two views aren't currently selected, so they are hidden.

This IS the currently selected view, so it's visible.

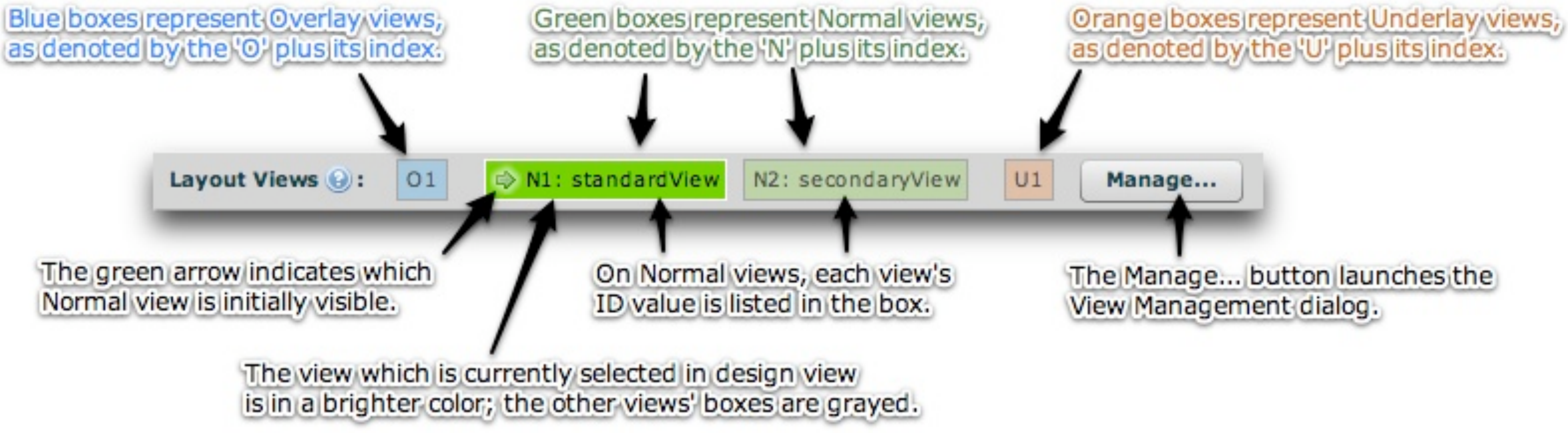
Underlay views; all are always visible. Stacked below all other views. Optional.

Managing Views in the Wizard

Within the Layout / Presentation section is the View Management Bar, located just below the Design Area:



Here is a diagram explaining the various parts of the management bar:

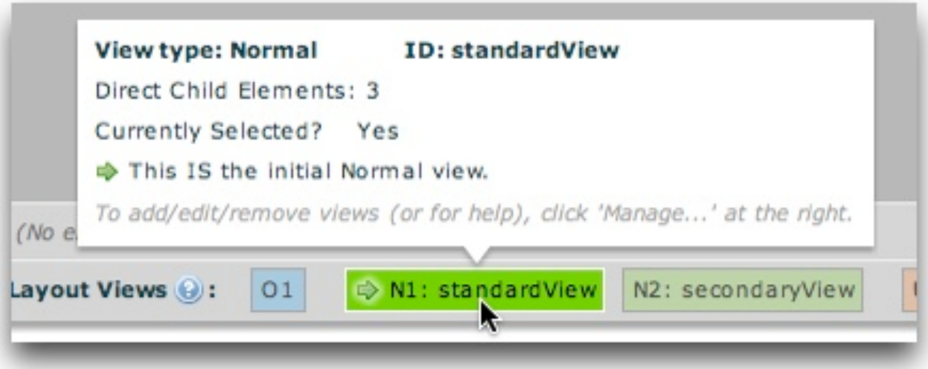

Blue boxes represent Overlay views, as denoted by the 'O' plus its index.
Green boxes represent Normal views, as denoted by the 'N' plus its index.
Orange boxes represent Underlay views, as denoted by the 'U' plus its index.
Layout Views
O1
N1: standardView
N2: secondaryView
U1
Manage...
The green arrow indicates which Normal view is initially visible.
On Normal views, each view's ID value is listed in the box.
The Manage... button launches the View Management dialog.
The view which is currently selected in design view is in a brighter color; the other views' boxes are grayed.

As noted in the photo, here is some information about the various aspects of the bar:

- There are several boxes displayed in the bar, one to represent each layout view which has been defined.
- Blue boxes represent Overlay views. The boxes for these views are labeled *O1*, *O2*, etc ('O' for Overlay).
- Green boxes represent Normal views. The boxes for these views are labeled *N1*, *N2*, etc ('N' for Normal), followed by the view's ID value.
- Orange boxes represent Underlay views. The boxes for these views are labeled *U1*, *U2*, etc ('U' for Underlay).
- On one of the Normal views, a green arrow will be shown. The arrow indicates that view is the initially displayed view.
- The 'Manage...' button on the right launches the View Management dialog where you can create, modify, and remove views.

To select a view for editing, click on its box once. Only one view is visible at a given time in design view to prevent confusion. The selected view is a brighter color in the bar; all other unselected views are grayed. To make changes.

Hovering over one of the view boxes will display more information about that view. Here is an example tooltip for a Normal view:



View type: Normal **ID: standardView**

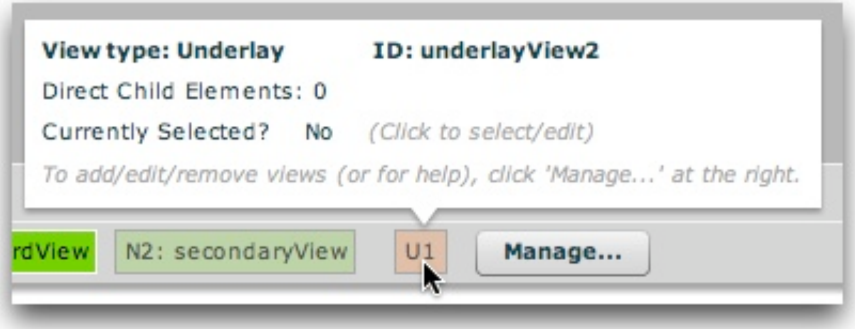
Direct Child Elements: 3

Currently Selected? Yes

➪ This IS the initial Normal view.

To add/edit/remove views (or for help), click 'Manage...' at the right.

And an example tooltip for an Underlay view:



View type: Underlay **ID: underlayView2**

Direct Child Elements: 0

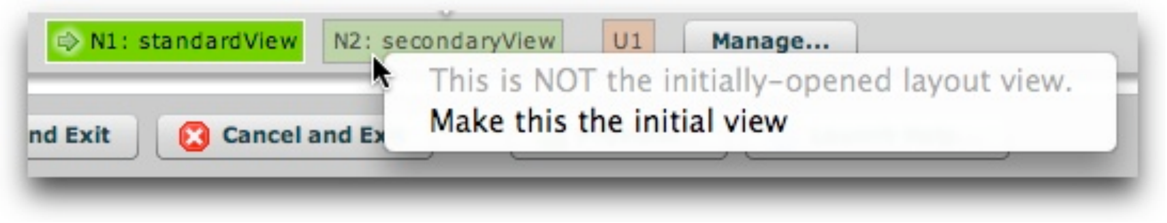
Currently Selected? No *(Click to select/edit)*

To add/edit/remove views (or for help), click 'Manage...' at the right.

Information in these tooltips includes:

- The view type
- The view's ID value
- The number of direct child elements in the view
- Whether the view is the currently selected view (for editing in the wizard). When a view is selected, it will be visible above in the Design Area.
- If the view is a Normal view, whether it is the initially displayed view.

Tip: If you have defined more than one Normal view, you can change which one is opened initially directly from the View Management Bar (without launching the View Management dialog). To do so, right-click on the view you want to be initial and select 'Make this the initial view':

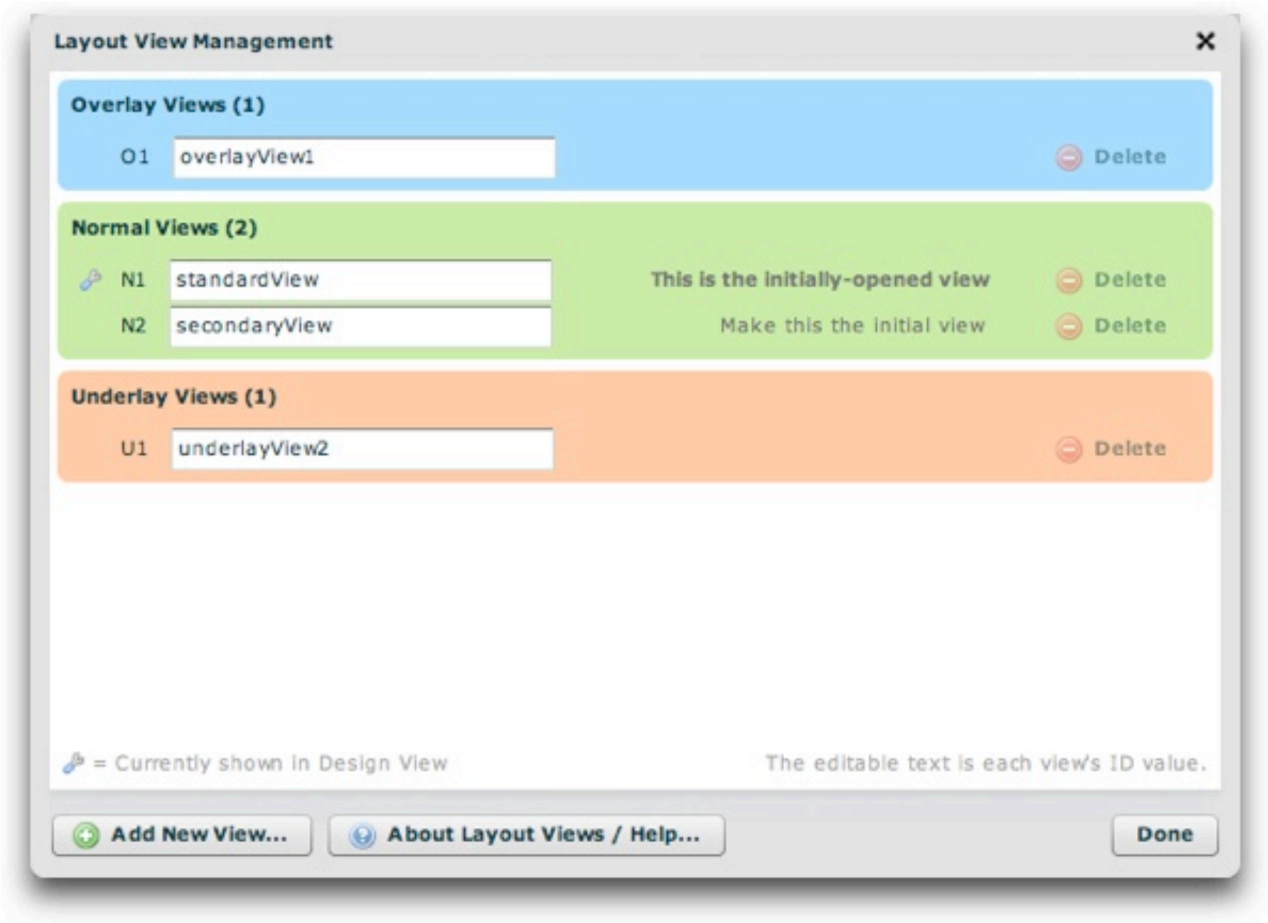


This is NOT the initially-opened layout view.

Tip - Understanding the 'Initially-Opened Layout View' setting: Only one Normal view is visible at a given time in the slideshow. When you create more than one Normal view, therefore, you must specify which one of these will be initially visible.

The View Management Dialog

Click the 'Manage...' button to launch the View Management Dialog. This is where you'll create, modify, and remove views. The dialog appears as follows:



Layout View Management

Overlay Views (1)

O1 overlayView1

Normal Views (2)

N1 standardView This is the initially-opened view

N2 secondaryView

Underlay Views (1)

U1 underlayView2

= Currently shown in Design View

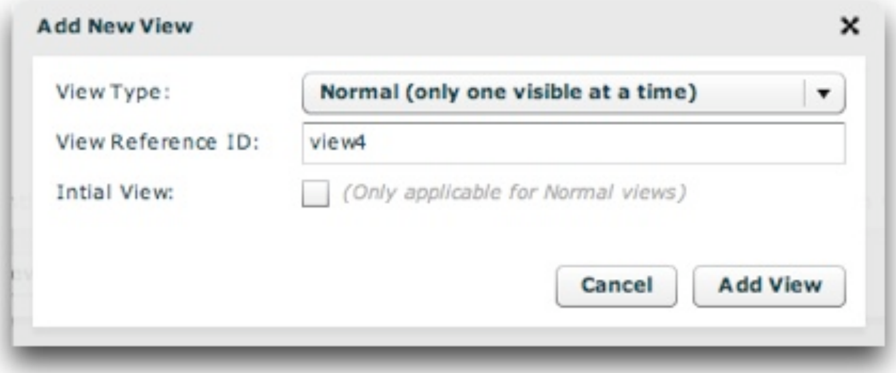
The editable text is each view's ID value.

A visual representation item is shown for each view, grouped by view type. The editable text box represents the view's ID value (only required for Normal views). This ID value is used to identify the view when changing between Normal views. Therefore, you should enter a meaningful name for each Normal view, and not modify it if you have referenced it elsewhere (e.g. in a LayoutJumpButton element).

The small wrench icon indicates which view is currently selected for editing in Design View. For normal views, you can change which is initially displayed by clicking the 'Make this the initial view' button next to the view you want initially displayed.

To add a new view, click the 'Add New View...' button:

The following prompt will open:



Add New View

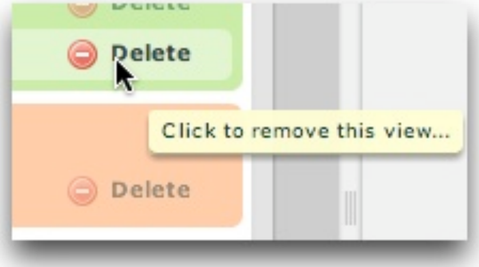
View Type: Normal (only one visible at a time)

View Reference ID: view4

Intial View: □ (Only applicable for Normal views)

You'll enter the view type (Normal, Overlay, or Underlay), reference ID, and -- if the view type is Normal -- whether the view is initially displayed. Then, click the Add View button to add the view.

To remove an existing view, click the Delete button on the right:



If any child elements have been added to that view, you will be asked to confirm the deletion.

For Advanced Users: Alternate Methods of Specifying the Initially Displayed View

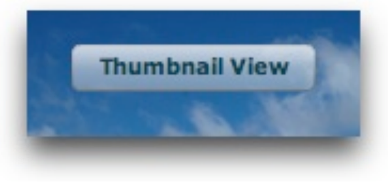
Note: This sub-section is only for advanced users and developers; beginners can skip over it.

You can specify the initial view by three different means, with each method overriding the methods listed before it.

- **1.** *(The default method)* By specifying the `<view ... initial="true">` attribute value. This is set visually in the wizard through the Manage Views window detailed above.
- **2.** By specifying a javascript callback in the slideshow's HTML page which defines the initial view id. For more information about this method, see the `XMLFlashSlideshow_v4_getInitialViewID` item in the `SlideshowInterface` class description. If specified, this value overrides the initial view ID specified by method one. (For more information, see the `com.dwuser.ss4.core.SlideshowInterface` class within the API Reference.)
- **3.** By specifying the `initViewID` parameter on the slideshow configuration object. This value is either passed in the slideshow's HTML configuration object (in a standalone implementation), or through the slideshow's `configurationParameters` object in the case of a component implementation. See the `FlashSlideshow.configurationParameters` property and the `FlashSlideshow` class description for more information. If specified, this value overrides the initial view ID specified by methods one and two. (For more information, see the `com.dwuser.ss4.ui.FlashSlideshow` class within the API Reference.)

Changing Between Views + View Transition Effects

At runtime (when your slideshow has been deployed), you can provide buttons to allow the user to switch between different Normal views (if you have defined more than one Normal view). This is done with the `LayoutJumpButton` element (in the Slideshow Control Buttons folder). This is a simple button which, when clicked, changes the current layout view. You will use the Properties Panel to specify the label for the button, as well as the view which will open when it is clicked. Here is an example of a layout jump button in action:



When the user transitions between views, a transition effect can be optionally played. These effects can be simple fades or complex ones such as a rotating cube. This is controlled in the wizard by using the **Layout Transition Settings** pod within the *Global Configuration and Playback Settings* section. See Using the 'Global Configuration and Playback Settings' Section for more information.

Note for Advanced Users & Developers: For more information about controlling the layout transition programmatically, see the `LayoutManager` class reference mentioned below in the 'Corresponding XML Structure' section.

Special View Parameters / Options

There are a number of special parameters which can specified for each view. These parameters must be passed through the XML, by adding attributes on the appropriate view node. To learn more about these options, see the `com.dwuser.ss4.ui.LayoutManager` class reference within the API Reference. Here is a brief overview of the various options which can be controlled:

- Stop Autoplay - A view can automatically stop/pause autoplay as soon as it is opened. This is often used to pause a slideshow while a user selects a different gallery.
- Start Autoplay - A view can automatically start/enable autoplay as soon as it is opened. This is often used to resume a paused slideshow once the user has selected a different gallery.
- Stop Audio - A view can be configured to automatically stop any audio playing on one or more levels (intro, gallery, and image). This is often used to pause audio while a user selects a different gallery.
- Start Audio - A view can be configured to automatically play any audio specified for one or more levels (intro, gallery, and image). This is often used to play audio once a user has selected a different gallery.
- Treat as Bitmap - Allows any non-embedded text within that view to be faded. Generally, for performance reasons, it is much better to add `treatAsBitmap` directly to individual elements, and not apply it to an entire view.

To learn more about the implementation of one of these specific options, see the reference mentioned above. Note that these modifications require familiarity editing XML, and are only recommended for advanced users and developers.

Corresponding XML Struture

For information about the XML structure behind layout views (and the user interface in general), see **The <ui> section - User Interface** within the Developer Reference. Also be sure to see the `com.dwuser.ss4.ui.LayoutManager` class reference within the API Reference.

Layout Element Overview

Once you've created the various layout views, you're ready to add content to each of them. As mentioned in the previous topic on views, you'll click on a view's box in the View Management Bar to select it for editing.

Each view contains various components. These components, also known as "elements", are small building blocks which are used to construct the user interface. For example, here are some of the available elements:

- The ImagesStandard element is an image-display element which displays the actual slides you have defined. If you were creating an images-only slideshow, this might be the only element you would use.
- The GalleryBar element is a compact bar which allows the user to select a gallery if you have added more than one gallery to the slideshow.
- The NextButton element is a button which, when clicked, progresses the slideshow to the next image/slide in the selected gallery.

Layout Containers / Nestable Components

In addition to the main elements which have a direct visual representation, there are also several non-visual layout container elements. These elements allow you to create fluid layouts which are easily resized and easier to manage and work with at design-time.

These layout elements are designed to hold child elements and lay them out in a specific way. The most common layout container elements are:

- **Canvas** - This is the container used by each view by default. It allows you to position elements using absolute coordinates, or constrain them to one or more sides of the container.
- **HoverCanvas** - This container is a special type of Canvas. Instead of always being visible, it automatically hides when the user is not active, then re-appears when they move their mouse again. This allows for hover-based controls, reducing visual clutter and removing distractions which prevent users from focusing on your presentation.
- **HBox** - This container lays out its child elements in a horizontal row, with each child directly next to its siblings. If one child gets larger, the other children automatically move to give it the necessary room. This element is commonly used for control bars, so that each button or control element is next to the previous one.
- **VBox** - This container is just like the HBox container, except it lays out its children vertically instead of horizontally.

Element Depth / Visual Stacking

When you add new elements to a layout, they are automatically positioned on top of other elements. To re-order elements visually, right-click on an item in the Design Area and select either Move Item Up (above other elements) or Move Item Down (below other elements). For more information, see the Design Area and Components Panel topic.

The depth of elements is controlled by their relative order in the Layout XML code. Elements listed later will appear on top of elements listed earlier in the XML. If you're comfortable with basic code editing, you can re-order elements by switching to code view and moving the highlighted node.

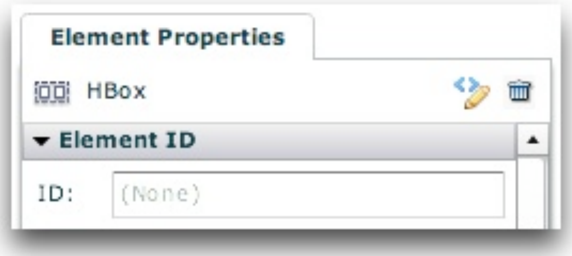
Element Positioning / Sizing and Styling

There are a number of different methods which can be used to position and size elements. These methods are covered in detail in the Element Positioning and Sizing topic.

All elements have numerous styling options which can be used to customize their appearance. This is covered in the upcoming Element Styling topic.

Referencing Elements Elsewhere in the Slideshow with ID Values

If you wish to reference a certain element elsewhere in your slideshow via code, you will need to specify a unique ID value for that element. In the wizard, this is done through the Element ID section of the Properties Panel:



ID: (None)

More Information for Advanced Users and Developers

More detailed information about the XML element structure and available attribute formats can be found in the reference for the `com.dwuser.ss4.ui.LayoutManager` class in the API Reference. Additionally, each component is represented by a class of the same name. Each of these classes is fully documented in the API Reference; each page lists all available properties and style settings for that element, as well as displaying an XML code example (helpful if editing the Layout XML directly).

Handling Element Events and Slideshow Events on Elements

Some elements dispatch events. The available events for each element are listed in the element's class documentation. Additionally, the main slideshow dispatches many events which notify of the slideshow status (see the `FlashSlideshow` class for a full listing). You can define handlers for these events by specifying an attribute on the element of this format: `event_eventNameHere="...handler code..."` (where the handler code is Actionscript). To add a handler which is triggered when a global slideshow event is dispatched, use the following format: `ssevent_eventNameHere="...handler code..."` (where the handler code is Actionscript). The following example shows a button whose label changes when it is clicked, and a label which displays a notification when the global `SlideshowEvent.LAYOUT_MANAGERREADY` event is dispatched:

```
<Button label="Not clicked yet..." event_click="event.target.label = 'Has been clicked!';" />
<Label id="sampleLabel" text="Waiting for the event to happen..."
ssevent_layoutManagerReady="sampleLabel.text += 'DONE!';" />
```

Element Positioning and Sizing

This topic covers the various means by which you can position and size elements in your layouts.

First, it is important to note that all numeric size and position values are denominated in pixels.

There are two different positioning situations: positioning within an absolute-position container and a relative-position container.

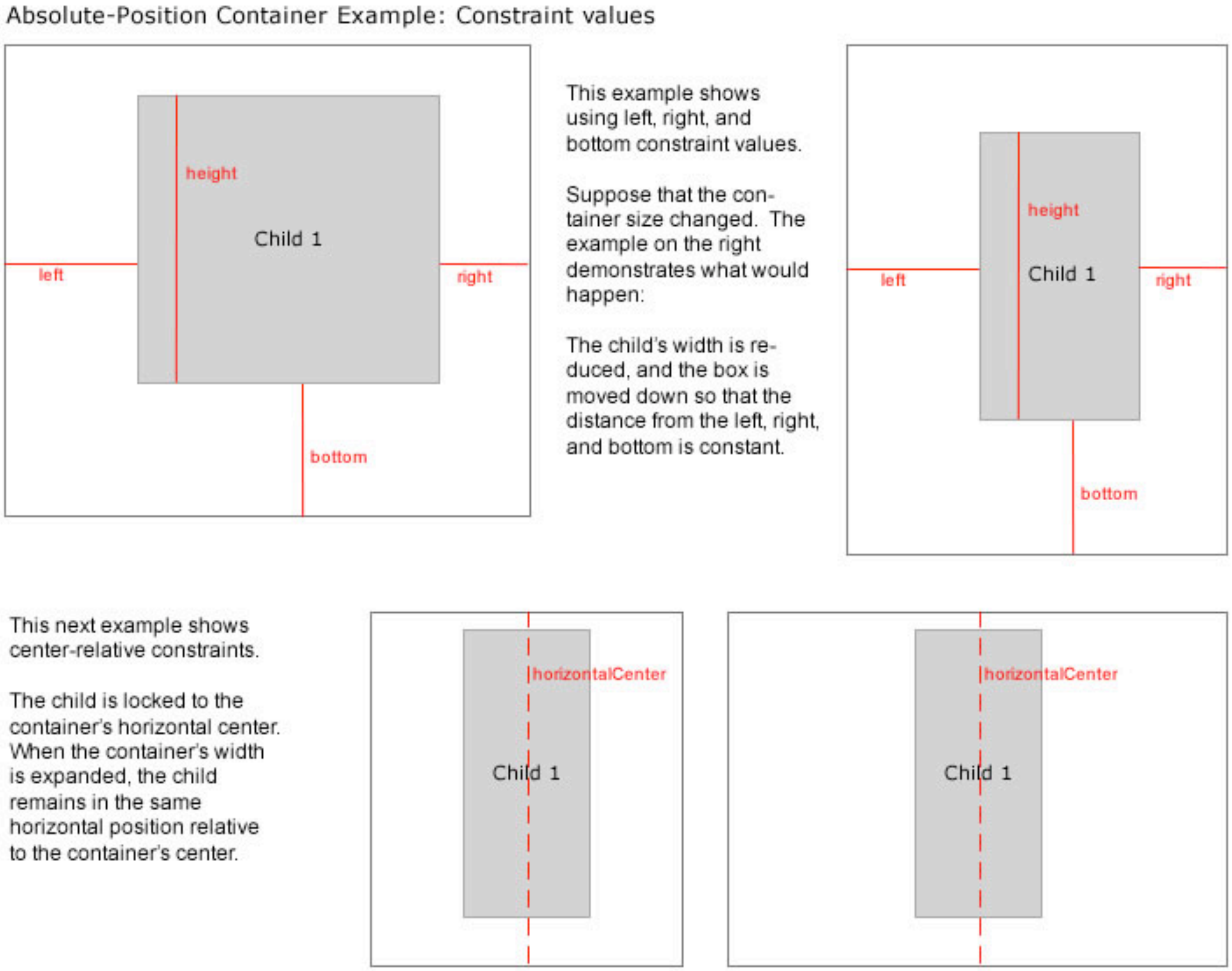
Using Elements within Absolute-Position Containers

Several containers implement absolute positioning. These include the Canvas and HoverCanvas containers, as well as the raw layout-view containers themselves (which are Canvas instances). In other words, when you drag a component directly onto the blank design area stage, it will be absolutely positioned. Note that all box containers (e.g. HBox, VBox, HFlowBox), the Panel, and TabNavigator do NOT implement absolute positioning; they use relative positioning. Absolute-position containers position each child in isolation without taking into account any of the other children.

Within absolute-position containers, you can use a combination of the following:

- **Constraint dimensions** - The `left`, `right`, `top` and `bottom` constraint properties can be used to anchor each side of a component to the corresponding edge of a container. The edges of the component stay a constant distance from the corresponding container edges, regardless of the container size. For example, you can specify a `left` value to lock the element to the left size of the container, and/or specify a `right` value to lock the element to the right edge of the container; these constraints will remain in effect even if the container changes size. If you specify constraints for both aspects of a given dimension (e.g. both `left` and `right`, or both `top` and `bottom`), the width or height will automatically be determined based on the space needed to be filled. Center-based constraints can also be used, positioning an element relative to the horizontal and/or vertical center of the parent container; this is done through the `horizontalCenter` and `verticalCenter` values. Because of the flexibility afforded when making changes later on, **constraints are the best way to position and size elements**.
- **X/Y Coordinates** - Similar to left and top values, `x` and `y` coordinates allow positioning of an element relative to the upper-left corner of the container.
- **Element Dimensions** - Explicit `width` and `height` values can be used, or you can allow the default element dimensions be used (default dimensions exist for most elements). Note that if both `left` and `right` constraint values are specified, this will function as a width value and override any explicit or default `width` setting. In the same way, specifying both `top` and `bottom` constraint values will override any `height` value. Supported values include **numeric pixel values** and **percentage values** (e.g. `100%`). If percentage-based dimensions exceed the boundaries of the parent container at runtime, they are automatically adjusted so that the element only goes up to (but not past) the edge of the container.

The following diagram illustrates the concept of constraint dimensions:



Absolute-Position Container Example: Constraint values

This example shows using left, right, and bottom constraint values.

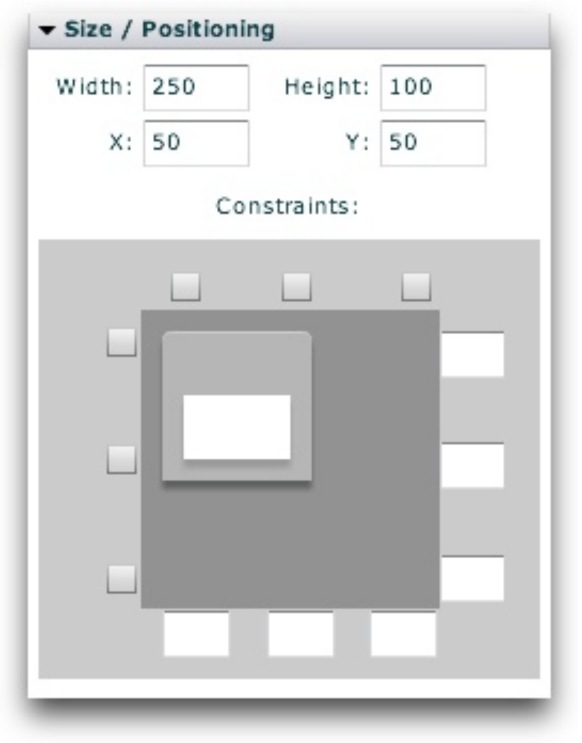
Suppose that the container size changed. The example on the right demonstrates what would happen:

The child's width is reduced, and the box is moved down so that the distance from the left, right, and bottom is constant.

This next example shows center-relative constraints.

The child is locked to the container's horizontal center. When the container's width is expanded, the child remains in the same horizontal position relative to the container's center.

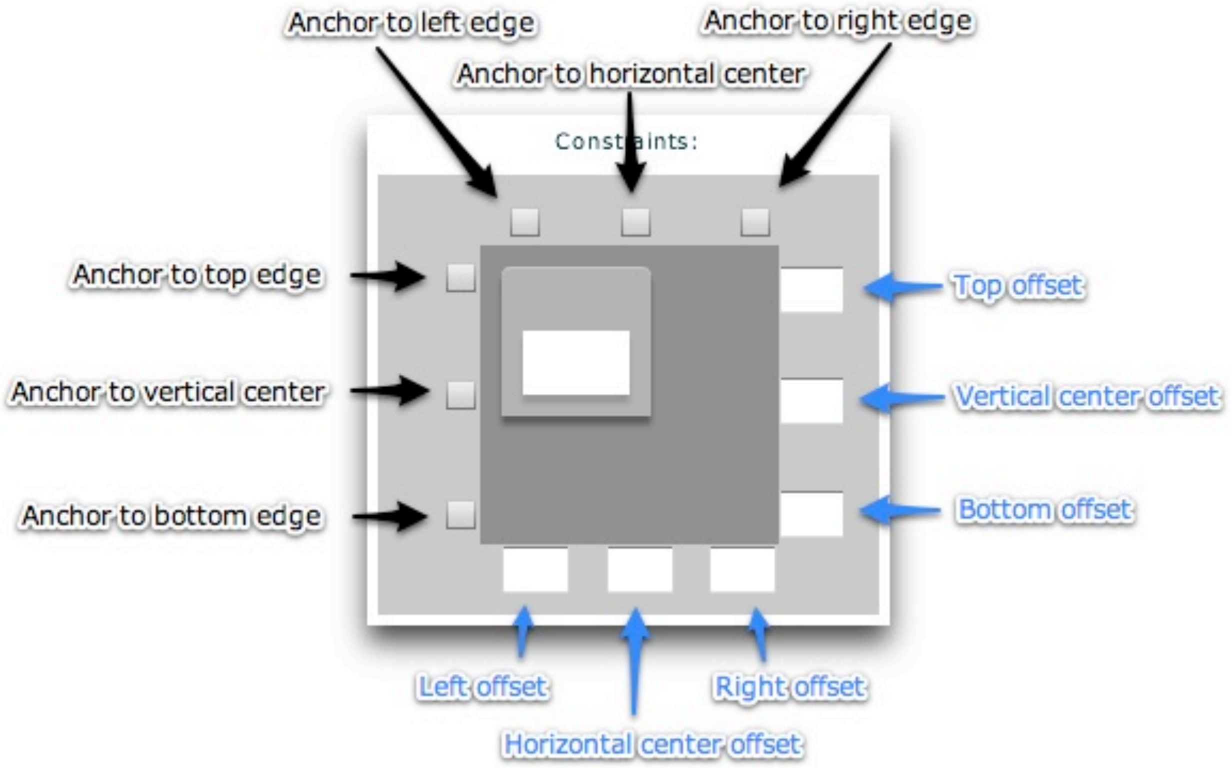
All positioning and sizing of elements in the wizard takes place in the Size / Positioning section of the Properties Panel. The section appears as follows for an element inside an absolute-position container:



Width: 250 Height: 100

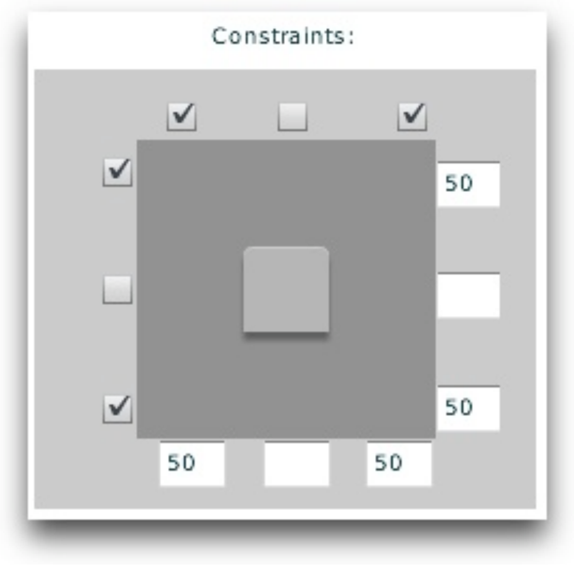
X: 50 Y: 50

The following diagram notes each of the constraint controls:

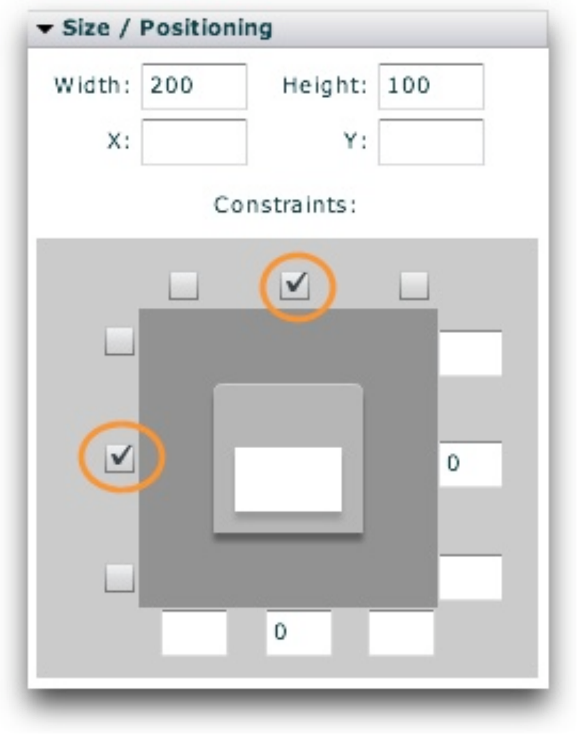


If you want to convert the current position into an edge-relative position without any math, just check the appropriate checkboxes. If, on the other hand, you know the exact pixel offsets you want, you can enter them directly in the corresponding offset boxes.

At the moment, only the x/y and width/height values are set on the example element shown in the diagram above. This makes the element 250x100 px, positioned at (50, 50). Suppose you wanted to anchor the example element above so that each of its sides where always 50 pixels from each edge. To do this, you would enter 50 in the top, bottom, left, and right offset fields. This automatically checks the corresponding anchor checkboxes:



Establishing these constraints would override any existing widht/height values which had been set. On the other hand, suppose you wanted the element to be 200 x 100 px, and always centered both vertically and horizontally. In this case, you would set the following:



Size / Positioning

Width: 200 Height: 100

X: Y:

Constraints:

Note: As you make changes, they are reflected in the Design Area.

Note: Removing the width or height values applied to an element makes the element revert to using its default dimensions if no dimension constraints are in effect.

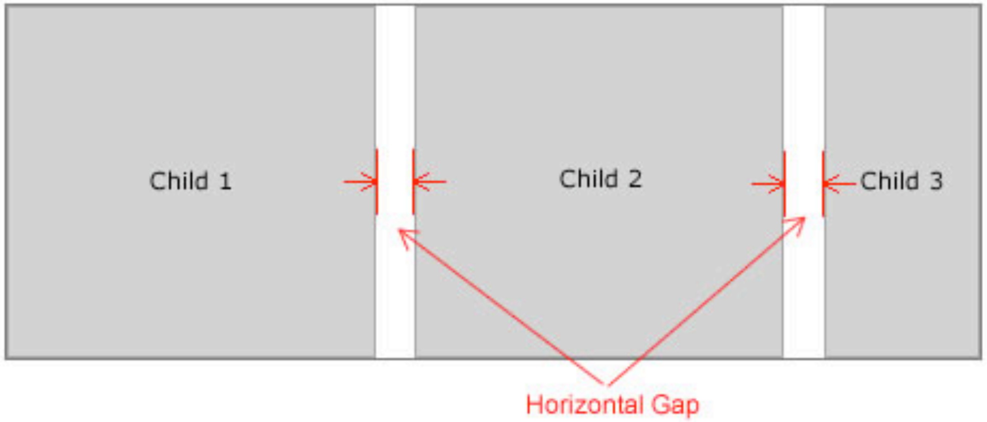
Using Elements within Relative-Position Containers

There are several layout containers which can be added to your layout that implement relative positioning. The most common of these are the box classes - HBox, VBox, HDividedBox and VDividedBox. Relative-position containers position each of their child components adjacent to each other based on the size of all children.

Within relative-position containers, **only Element Dimensions can be used**. All positioning is automatically determined by the container, based on the size and position of other sibling element(s). Explicit `width` and `height` values can be used, or you can allow the default element dimensions be used (default dimensions exist for most elements). Supported values include **numeric pixel values** and **percentage values** (e.g. `100%`). If percentage-based dimensions exceed the boundaries of the parent container at runtime, they are automatically adjusted so that the element only goes up to (but not past) the edge of the container.

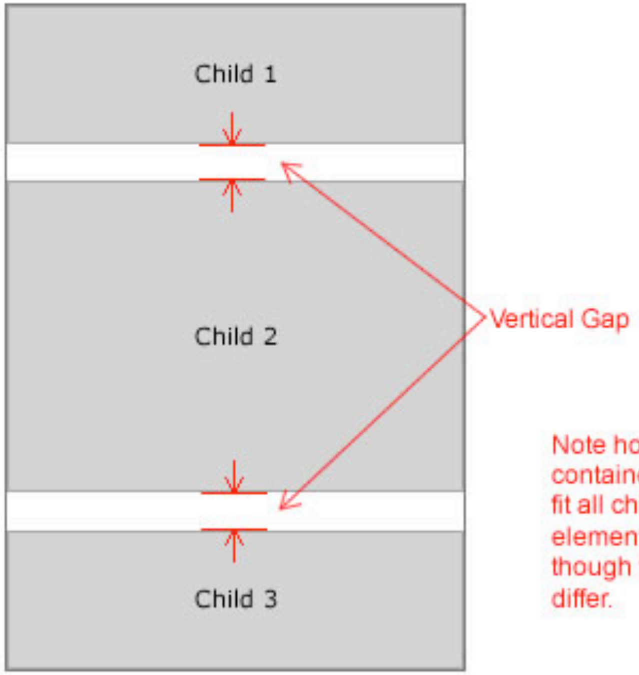
The following diagram illustrates the concept of two different relative-position containers -- the HBox and VBox, the most commonly used containers:

Relative-Position Container Example: **Hbox**



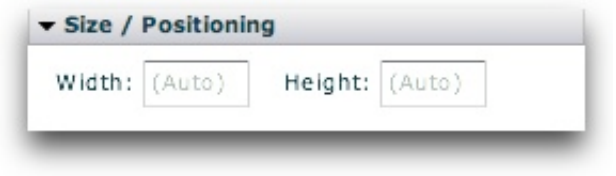
Note how the container sizes to fit all child elements, even though their sizes differ.

Relative-Position Container Example: **Vbox**



Note how the container sizes to fit all child elements, even though their sizes differ.

All positioning and sizing of elements in the wizard takes place in the Size / Positioning section of the Properties Panel. The section appears as follows for an element inside a relative-position container; it's simpler than the absoute-positioned counterpart:



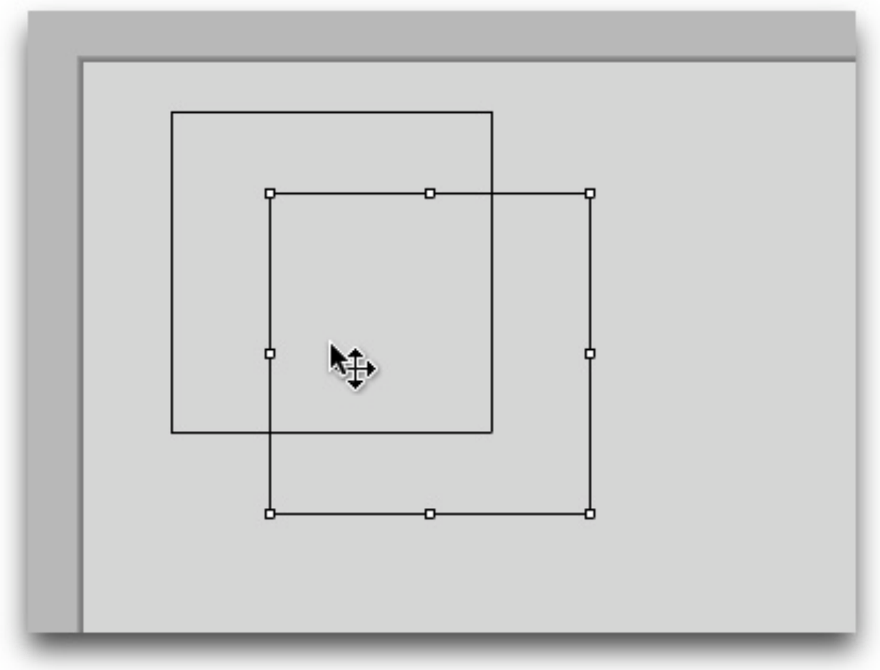
▾ Size / Positioning

Width: (Auto) Height: (Auto)

As you can see, there are only two settings: the height and the width. If you leave either blank, the value will be automatically determined for most elements. Additionally, removing the existing width or height values applied to an element makes the element revert to using its default dimensions.

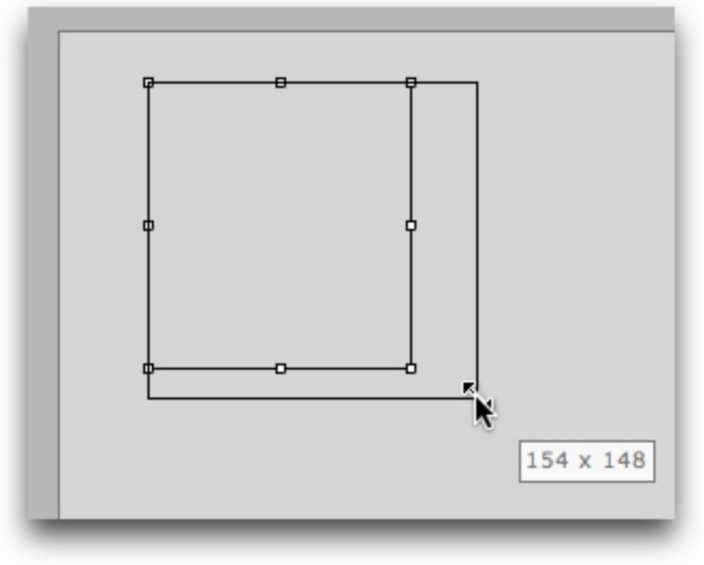
Resizing and Repositioning Directly in Design View

In the wizard, you can also resize and reposition automatically by dragging. To reposition an element, click and drag in the middle of it:



A 'move' cursor will appear. You can drag and drop elements from one container to another. If constraint positioning or dimensions have been established for the element, these constraints will be retained and only the offset values adjusted. On the other hand, if x/y positioning coordinates were used, these values will be updated directly. In other words, the wizard is intelligent and won't overwrite your positioning or sizing work.

To resize an element, click it once to select it. On each corner (and on the middle of each side), a resize handle will appear. When you hover over one of these handles, a resize cursor will appear. If you click and drag, the element size will be updated as appropriate. A tooltip is shown while dragging to indicate the new effective size of the element:



As with dragging to move an element, all constraint positioning and/or constraint-based dimensions are intelligently respected and retained.

Element Styling

There are many options available to customize the appearance and behavior of the layout elements which make up a slideshow's user interface. These options are collectively known as **styles**.

Four different types of styles exist; each are covered in more detail below.

Element-Specific Styles (AKA Instance-Specific Styles)

These styles are applied directly to a single element instance using the Properties Panel in the wizard (or by adding attributes directly to the element's XML node if editing the XML directly). Because they are instance-specific, they have no effect on any other instances of the element.

Important Note: The common style options for each element can be accessed directly in the wizard by using the Properties Panel. To view a comprehensive list of *all* styles and properties available for each element, view the class reference for the element in question in the API Reference.

Important Note: Properties are **always** instance-specific. They are not inherited, nor can they be defined globally. Only style properties (as listed under the Styles section in the API reference) can be inherited or defined globally.

Inherited Styles

Some style properties automatically are passed from a parent element to all child instances. The most common examples are:

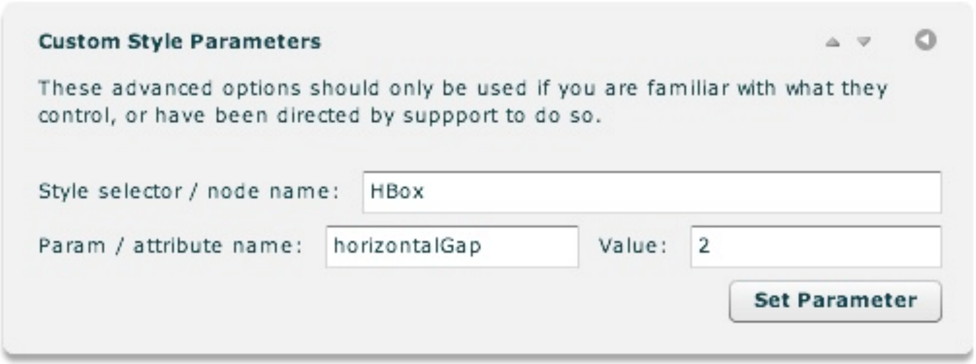
- **Theme Color / The Global Theme Color** - The Global Theme Color setting is set in the 'Global Theme Color & Default Font' pod in the main Appearance Settings panel. This value provides the global default for many colors, especially control button theme colors. If you explicitly define a different Theme Color for a certain element, all of that element's child elements will inherit that value.
- **Font Settings / Default Font Styles** - Font settings are inherited as well. The Default Font Styles specified in the 'Global Theme Color & Default Font' pod in the main Appearance Settings panel provide defaults for all elements with text throughout the slideshow. You can customize the appearance of a specific element's text (and the text of any of its child elements, if it holds child elements) by changing the values in the Text Formatting section of the Properties Panel.

Global Element Styles with a Direct Visual Representation

If you wish to customize a non-inheriting style on all instances of a certain element without repeatedly setting a value, you can use global element styles.

These styles are defined in the `<styles>` section of the XML configuration file, using a selector (node name) equal to the element's name. For example, `<styles> ... <ImagesStandard someProperty="someValue" /> ... </styles>` would set the `someProperty` style/property to `someValue` on all ImagesStandard element instances, unless it were overridden on a specific instance with an instance-specific style value.

To apply global element styles in the wizard, you will use the 'Custom Style Parameters' pod in the main Global Configuration and Playback Settings tab:



Custom Style Parameters

These advanced options should only be used if you are familiar with what they control, or have been directed by suppport to do so.

Style selector / node name: HBox

Param / attribute name: horizontalGap Value: 2

First, you enter the Style Selector. Then, you enter the parameter and its value. In the example shown above, a default value of `2` is being established for the `horizontalGap` property on all HBox elements.

As mentioned above, to view a comprehensive list of *all* styles available for each element, view the class reference for the element in question in the API Reference.

Global Styles / Settings without a Direct Visual Representation

The preceding section addressed global styles for elements which visibly are defined in your slideshow layout. However, there are also other styles which have no direct visual representation. This encompasses both special element states as well as non-visual settings which are set by using style values.

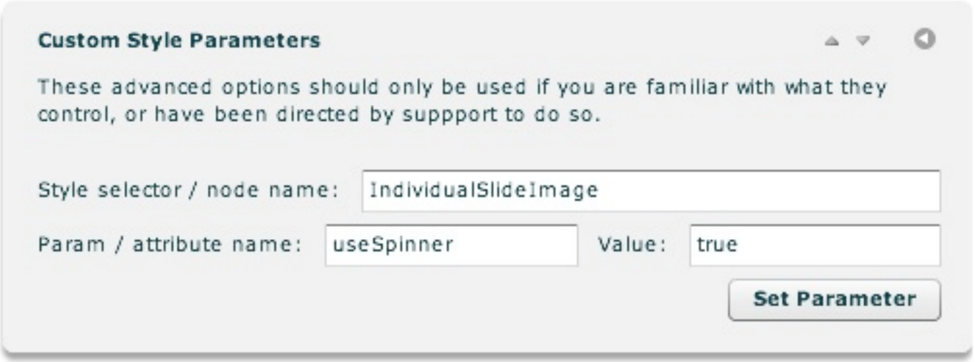
One of the most common examples is the Thumbnails element. For each Thumb instance in the Thumbnails element, there are three different states:

- The **Normal** state, when the thumbnail has not been modified in any way and is sitting there waiting for user interaction.
- The **Hover** state, when the user hovers over the thumbnail.
- The **Selected** state, when the thumb represents the currently selected image.
- The **Visited** state, when the thumb represents an image which has been viewed.

Even though none of these states visibly exist in your layout, you still want to be able to apply styles to them in order to customize each state. For example, you might want to add a border to a hovered-over thumbnail, or gray out visited thumbnails. (For this example, the mentioned settings can be visually edited through the 'Common / Layout & Appearance' section of the Properties Panel when a Thumbnails instance is selected.)

As another common use, there are many generic settings which are specified through special global style selectors. For instance, the IndividualSlide and IndividualSlideImage selectors provide control over the appearance of each of the slides in a slideshow. The ImageManager selector provides control over many generic settings such as the loading method used. (Many of the styles/settings mentioned here can be edited visually in the wizard, through the Slide Appearance and Global Configuration & Playback Settings main tabs. Details on all available settings, along with XML examples, can be found on the various class files in the API Reference.)

To make customizations such as this for non-visual styles and settings, you use a global style selector, in the same manner detailed in the preceding 'Global Element Styles with a Direct Visual Representation' section. As mentioned in that section, to apply global styles / settings in the wizard, you will use the 'Custom Style Parameters' pod in the main Global Configuration and Playback Settings tab:



Custom Style Parameters

These advanced options should only be used if you are familiar with what they control, or have been directed by suppport to do so.

Style selector / node name: IndividualSlideImage

Param / attribute name: useSpinner Value: true

You enter a selector, parameter name, and value. In this example, the non-visual IndividualSlideImage selector is used. The useSpinner property is being enabled.

As mentioned above, details on all available selectors and their settings, along with XML examples, can be found in the class files in the API Reference.

Using the External Wizard to Create and Edit Slideshows > Section-by-Section Usage Guide > Using the 'Layout / Presentation' Section > **Available Elements (incl. Cheatsheet)**

Available Elements (including Cheatsheet)

This section covers in detail all of the available user interface elements which can be used to build slideshows. They are grouped in the same categories as used in the Components Panel in the wizard, with each group in a separate subtopic.

While great detail is included in this section of the wizard, even more information on every component is available in the API reference, under the 'Equivalent Class File' listed for each element.

Usage Tip: Many of the commonly edited parameters accessible in the Properties Panel are noted. However, not all are included. For *all* parameters in the Properties Panel, you can hover over their names for a detailed description.

Printable Cheatsheet

A quick and printable reference cheatsheet summarizing all elements is shown in the following table:

Element Name	Quick Description
	Layout Containers - Used to hold and arrange other elements
HFlowBox	[Rarely used] Same as HBox, except wraps child elements to multiple rows if needed.
HDividedBox	[Rarely used] A relative-position horizontal container, with each child resizable by the user.
VDividedBox	[Rarely used] A relative-position vertical container, with each child resizable by the user.
HBox	A relative-position horizontal container, with each child positioned adjacent to the next. Useful for control bars.
VBox	A relative-position vertical container, with each child position adjacent to the next. Useful for vertical control bars.
HoverCanvas	A special absolute-position layout container which automatically hides when the user stops interacting with it, and re-appears when the user moves the mouse again.
HoverSlideCanvas	A special type of HoverCanvas (see directly above) that hides itself by sliding out of view to one side or another. It only differs from the HoverCanvas in its special built-in hide/show effect.
Canvas	An absolute-position layout container. Used to hold/organize a group elements.
Panel	[Rarely used] A "pod"-style layout container with a title, useful for segmenting a user interface.

Tile	[Rarely used] A layout container which lays its children out in a grid.
TabNavigator	[Rarely used] A layout container with one child visible at a time, with each accessible by a tab.
Image Progress Loaders - Display loading progress	
Centered Loaders Block	Special starter element which contains all loaders, centered and ready to be dropped into an Overlay layout view.
SingleLoader	Displays the progress of loading when a single image is being loaded.
GroupLoader	Display the progress of loading when a group of images are being loaded.
Slideshow Image Displays - Displays the images/slides in the slideshow	
ImagesStandard	The most common image display element. Displays all slides in the standard slideshow view, with transitions between each slide.
ImagesScrollList	Displays all slides in a contiguous row, column, or grid. Allows multiple slides to be visible at a given time.
Images3DWall	[Pro version only] Display all slides on a virtual 3D wall or carousel or wave; slides can be brought into focus to be viewed up close.
ImagesCoverFlow	[Pro version only] Displays all slides in an iTunes-style coverflow.
ImagesStarfield	[Pro version only] Display all slides in a virtual 3D starfield.
ImagesScatter	[Pro version only] Displays all slides scatterd across a virtual table. A single slide can be brought up from the table to be viewed up close.
Main Slideshow Elements - Essential slideshow elements	
Pre-Built Hover-only Control Bar - NumberBar	This special starter element is a horizontal control bar which contains a NumberBar; it is only visible when the user moves their mouse. It is designed to go at the bottom of the slideshow, but may be moved anywhere. The default buttons can be customized as needed.
Pre-Built Hover-only Control Bar - Thumbnails	This special starter element is a horizontal control bar which displays a row of thumbnails; it is only visible when the user moves their mouse. It is designed to go at the bottom of the slideshow, but may be moved anywhere. The default buttons can be customized as needed.
Pre-Built Control Bar - NumberBar	This special starter element is a horizontal control bar which contains a NumberBar; it is always visible, regardless of interaction. It is designed to go at the bottom of the slideshow, but may be moved anywhere. The default buttons can be customized as needed.
Pre-Built Control Bar - Thumbnails	This special starter element is a horizontal control bar which displays a row of thumbnails; it is always visible, regardless of interaction. It is designed to go at the bottom of the slideshow, but may be moved anywhere. The default buttons can be customized as needed.
ImageTree	[Rarely used] Displays all galleries and images in a hierarchical tree which can be used for navigation.
Thumbnails	Displays the thumbnails for all images in the currently-selected gallery. Analogous to the v3 "ThumbBar".
NumberBar	Displays a row, column, or grid of numbers, with each one representing a slide in the currently-selected gallery. Hovering

	over a number provides a tooltip with more image information. Analagous to the v3 "JumpBar".
GallerySelector	Displays a grid of all galleries, allowing for selection of a gallery. Analogous to the v3 "Gallery" element.
GalleryBar	A compact bar which displays the title of all galleries, allowing selection of a gallery. Hovering over a gallery's title displays its description and thumbnail.
Caption	Displays the current image's title and caption in a box which can be toggled on and off via the CaptionToggleButton element.
DynamicScrollableText	Displays scrollable text which changes based on the current image, providing image-specific information. Can be used to display image title, caption, etc.
DynamicLinkButton	A link button which can be clicked to perform some action (such as going to a URL). The label and link can change based on the current image, providing image-specific information.
DynamicText	Displays NON-scrollable text which changes based on the current image, providing image-specific information. Can be used to display image title, caption, etc.
Slideshow Control Buttons - Provide control over the slideshow	
AutoplayLoopToggleButton	A checkbox which allows the user to enable or disable autoplay looping.
LastButton	A button which causes the slideshow to jump to the final image in the gallery.
PlayPauseButton	Plays/pauses the slideshow. Controls autoplay, audio, and video.
FirstButton	A button which causes the slideshow to jump to the first image in the gallery.
PreviousButton	A button which causes the slideshow to jump to the previous image.
LayoutJumpButton	A button which can be used to trigger a change between various layout views.
NextButton	A button which causes the slideshow to jump to the next image.
AutoplayToggleButton	A play/pause button which only controls autoplay (not audio or video).
FullScreenButton	Allows the user to toggle in and out of fullscreen mode.
CaptionToggleButton	Allows the user to hide or show any Caption elements which have been added in the slideshow.
PrintButton	Allows the user to print the current slide.
Slideshow Option Controls - Provide control over various slideshow options	
AutoplayDelaySlider	Allows the user to control the delay between each image when autoplay is enabled.
AutoplayProgressDisplay	Displays an indicator of how long until autoplay causes the slideshow to automatically progress to the next image.
GotoBox	Allows the user to type in the index of a number and jump directly to it.
PagingControlSizer	Allows the user to specify the size of each tile in a Thumbnails or NumberBar element.
Extras / Decorations - For enhancing the slideshow user interface	
HRule	Provides a horizontal line, often used as a separator between two children in a vertical relative-layout container.

VRule	Provides a vertical line, often used as a separator between two children in a horizontal relative-layout container.
BorderBox	A box which acts as a background element, with background, border, and/or drop shadow. Can be used to display the current slide's dominant color.
Spacer	Used in relative-layout containers to provide spacing between elements.
Audio Control Elements - Provide control audio	
AudioPlayPauseButton	Plays/pauses audio on one or more levels (background, gallery-specific, image-specific).
AudioControlCanvas	A special holder element which should be used for holding audio control elements to ensure that they control the appropriate audio level(s).
AudioScrubBar	Allows the user to scrub the position of audio on one or more levels (background, gallery-specific, image-specific).
AudioMuteButtons	Allows the user to mute all audio.
AudioControlBar	A basic audio controller; can be used to control audio on one or more levels (background, gallery-specific, image-specific).
AudioLoudButton	Allows the user to un-mute all audio, making it full-volume.
AudioVolumeSlider	A slider which allows the user to adjust the volume of all audio.
AudioVolumeToggleButton	Allows the user to toggle between muting/unmuting all audio.
AudioVolumeControl	Provies a mute button, a volume slider, and a loud (full-volume) button to control the volume all audio.
AudioLoadBar	Displays the loading progress of audio on one or more levels (background, gallery-specific, image-specific).
AudioInfoText	Display text with information specific to currently playing audio on a certain level (background, gallery-specific, image-specific).
Static Generic Elements - Generic user interface assets which don't change	
Button	A generic button which has no default action; can have custom actions applied to it.
LinkButton	A generic link button which has no default action; can have custom actions applied to it.
Image	A non-changing image, used for items such as logos or background images.
Text	Displays non-changing text; can be multiple lines (whereas Label cannot).
Label	Displays non-changing one-line text.
FlexVideoPlayer	[Rarely used] Displays a static video. It is unrelated to the video associated with a slide, and is rarely applicable.
Special Starter Elements - Helpful building blocks with preset configurations	
Dominant-slide-color BorderBox	A BorderBox which is configured to automatically display the current slide's dominant color. Can be used to make the slideshow background match the current slide.
Plain Title Text	A dynamic text box which displays the current image's title and can be fully styled.
Plain Caption Text	A dynamic text box which displays the current image's description / caption and can be fully styled.
Centered Loaders Block	Special starter element which contains all loading indicators, centered and ready to be dropped into an Overlay layout view.

Pre-Built Hover-only Control Bar - NumberBar	This special starter element is a horizontal control bar which contains a NumberBar; it is only visible when the user moves their mouse. It is designed to go at the bottom of the slideshow, but may be moved anywhere. The default buttons can be customized as needed.
Pre-Built Hover-only Control Bar - Thumbnails	This special starter element is a horizontal control bar which displays a row of thumbnails; it is only visible when the user moves their mouse. It is designed to go at the bottom of the slideshow, but may be moved anywhere. The default buttons can be customized as needed.
Pre-Built Control Bar - NumberBar	This special starter element is a horizontal control bar which contains a NumberBar; it is always visible, regardless of interaction. It is designed to go at the bottom of the slideshow, but may be moved anywhere. The default buttons can be customized as needed.
Pre-Built Control Bar - Thumbnails	This special starter element is a horizontal control bar which displays a row of thumbnails; it is always visible, regardless of interaction. It is designed to go at the bottom of the slideshow, but may be moved anywhere. The default buttons can be customized as needed.

Layout Containers

Layout containers are a class of elements which are used when building layouts to hold and arrange one or more child elements.

Elements in this section:

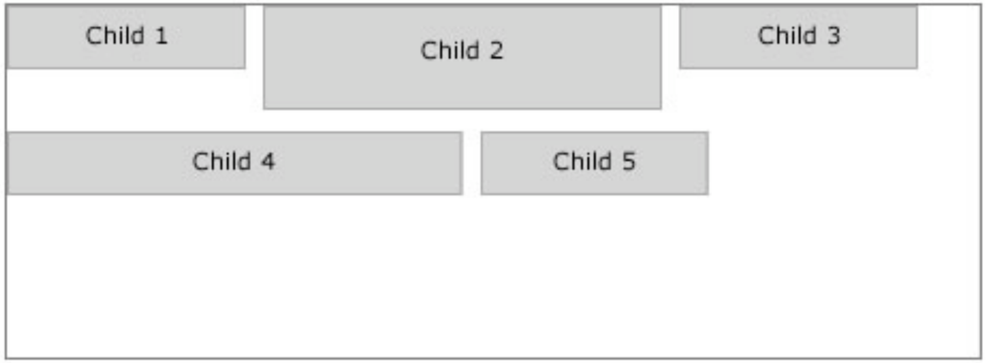
- HFlowBox
- HDividedBox
- VBox
- TabNavigator
- HBox
- VDividedBox
- Tile
- Panel
- HoverCanvas
- Canvas

Noteworthy Special Element for Developers

While it has no visual representation, it is important to note the DataLooper element, which is used to create multiple copies of a component based on a list of data. This component, implemented by the `com.dwuser.ss4.ui.DataLooper` class (see the API Reference), can be helpful for developers who are making code-based customizations.

HFlowBox

The HFloxBox element lays out its child elements in a horizontal row until the available width has been consumed. If more children still need to be laid out, it wraps to a new row and continues laying out the child elements. The following diagram illustrates the way its child elements are laid out:



Important Usage Notes

Usually the HBox container is a better choice than this one. Due to the small initial design-view height rendering, it can be difficult to drag elements into it. To work around this, it is easiest to switch to Source view and move the other element(s) so that they are HFlowBox child nodes (within `<HFlowBox ... > ... </HFlowBox>`).

Noteworthy Parameters

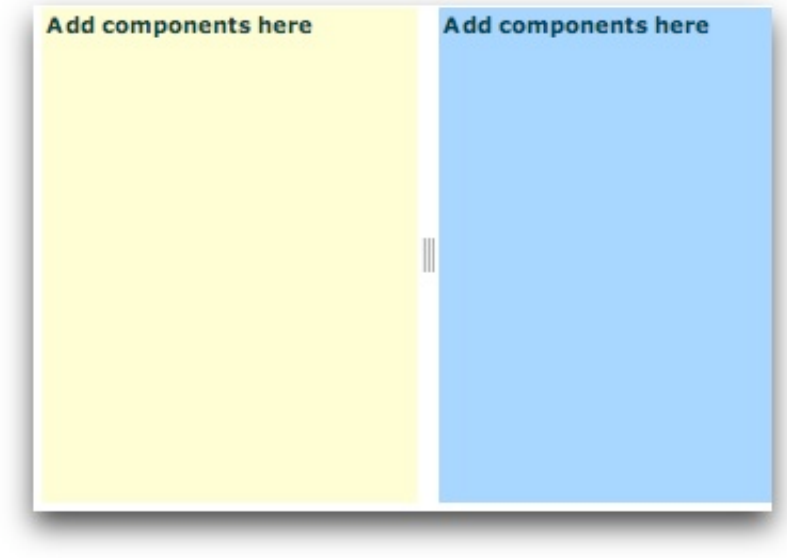
Parameter name in Properties Panel	XML Property Name	Notes
Horizontal Gap	`horizontalGap`	This specifies, in pixels, the horizontal spacing between each of the child elements.
Vertical Gap	`verticalGap`	This specifies, in pixels, the vertical spacing between each row of child elements.

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.containers.HFlowBox` class in the API Reference.

HDividedBox

The HDividedBox is essentially the same as the HBox except that each child can be resized by the user. Between each child element, a draggable divider is shown:



Important Usage Notes

The HBox container is often a better choice than this one; the benefit of this container is that it allows the user to make size adjustments on-the-fly.

Noteworthy Parameters

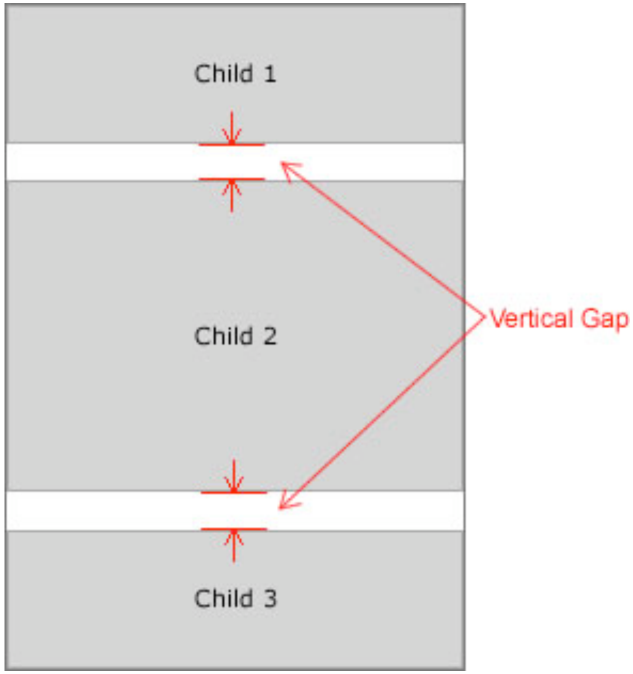
(No noteworthy specific parameters)

Equivalement Class File

For detailed reference on this element, see the `mx.containers.HDividedBox` class in the API Reference.

VBox

The VBox container arranges its children adjacent to each other in a vertical column. This container is often used to build layouts where the child elements need to be next to each other. The following diagram illustrates how the layout works:



Important Usage Notes

After adding the first child element, you can remove the default dimensions (width / height values) from the box in the Properties Panel. This will allow the VBox to expand as needed to accomodate its child elements.

As with all containers, you can add a background, border and/or drop shadow to this container using the Border / Fill and Drop Shadow sections of the Properties Panel.

Noteworthy Parameters

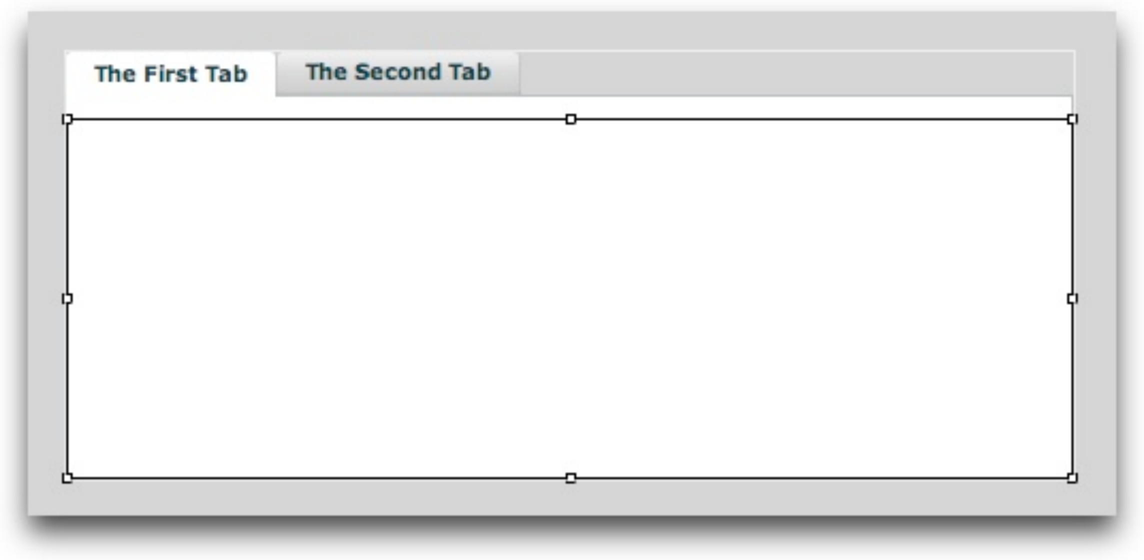
Parameter name in Properties Panel	XML Property Name	Notes
Vertical Gap	`verticalGap`	This specifies, in pixels, the vertical spacing between the child elements.
Horizontal Align	`horizontalAlign`	When the widths of the child elements vary, this specifies the horizontal alignment of the narrower elements within the container.
Vertical Align	`verticalAlign`	If you have specified an explicit height for the VBox and the content does not fill the full height, this specifies the vertical alignment of all elements within the container.

Equivalement Class File

For detailed reference on this element, see the `mx.containers.VBox` class in the API Reference.

TabNavigator

The TabNavigator allows you to create a tabbed interface within the slideshow. Below is an example of a TabNavigator in Design View; two empty sub-containers have been added to it:



Important Usage Notes

To make a TabNavigator function, you must add a sub-container to it for each tab you wish to create. Usually, this will be a Canvas, HBox, or VBox element. To add a sub-container to a TabNavigator, drag and drop it onto the top of the container (where the tabs appear). For each sub-container, the **Label** property in the Properties Panel (the `label` attribute in the XML) is used by the TabNavigator as that tab's label.

Each sub-container which is added should have its width and height values set to `100%` to ensure that it fully fills the TabNavigator.

In the example above, two VBox elements were dragged-and-dropped onto the TabNavigator. Both had their Width and Height settings set to `100%`. The Label property on the first VBox was set to 'The First Tab', and the Label property on the second VBox was set to 'The Second Tab.'

Noteworthy Parameters

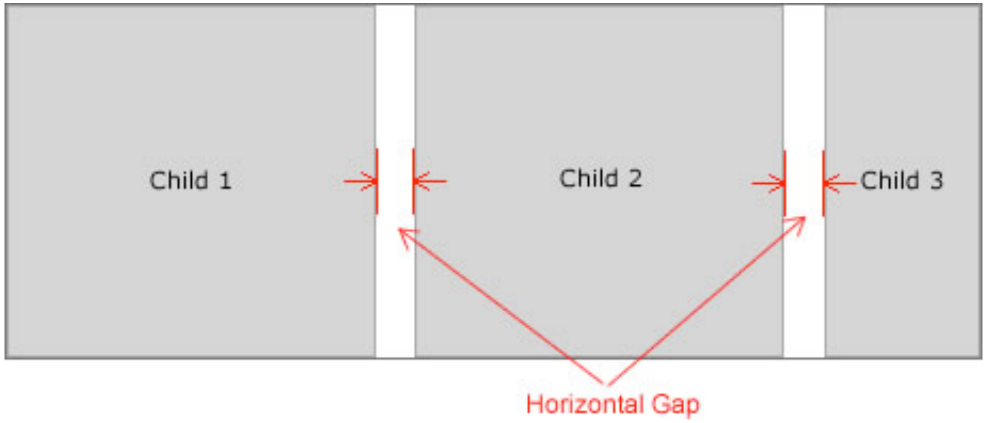
(No noteworthy specific parameters)

Equivalement Class File

For detailed reference on this element, see the `mx.containers.TabNavigator` class in the API Reference.

HBox

The HBox container arranges its children adjacent to each other in a horizontal column. This container is often used to build layouts where the child elements need to be next to each other, especially horizontal control bars. The following diagram illustrates how the layout works:



Important Usage Notes

After adding the first child element, you can remove the default dimensions (width / height values) from the box in the Properties Panel. This will allow the HBox to expand as needed to accomodate its child elements.

As with all containers, you can add a background, border and/or drop shadow to this container using the Border / Fill and Drop Shadow sections of the Properties Panel.

Noteworthy Parameters

Parameter name in	XML Property	Notes

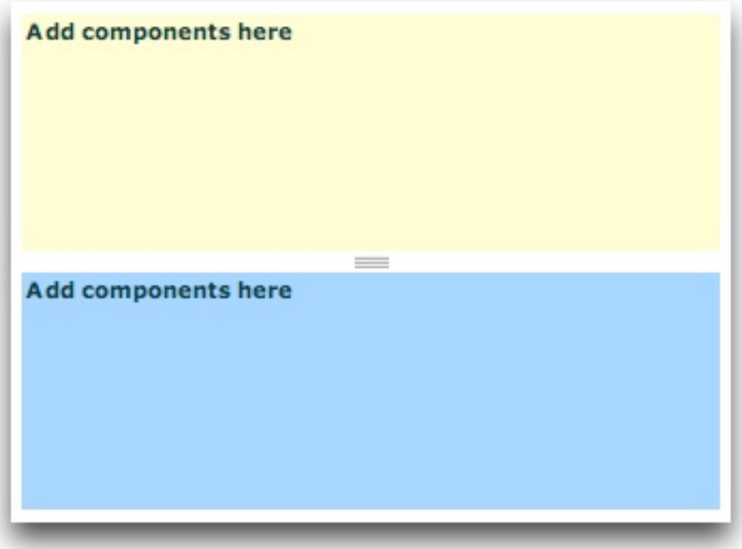
Properties Panel	Name	
Horizontal Gap	`horizontalGap`	This specifies, in pixels, the horizontal spacing between the child elements.
Horizontal Align	`horizontalAlign`	If you have specified an explicit width for the HBox and the content does not fill the full width, this specifies the horizontal alignment of all elements within the container.
Vertical Align	`verticalAlign`	When the heights of the child elements vary, this specifies the vertical alignment of the shorter elements within the container.

Equivalement Class File

For detailed reference on this element, see the `mx.containers.HBox` class in the API Reference.

VDividedBox

The VDividedBox is essentially the same as the VBox except that each child can be resized by the user. Between each child element, a draggable divider is shown:



Important Usage Notes

The VBox container is often a better choice than this one; the benefit of this container is that it allows the user to make size adjustments on-the-fly.

Noteworthy Parameters

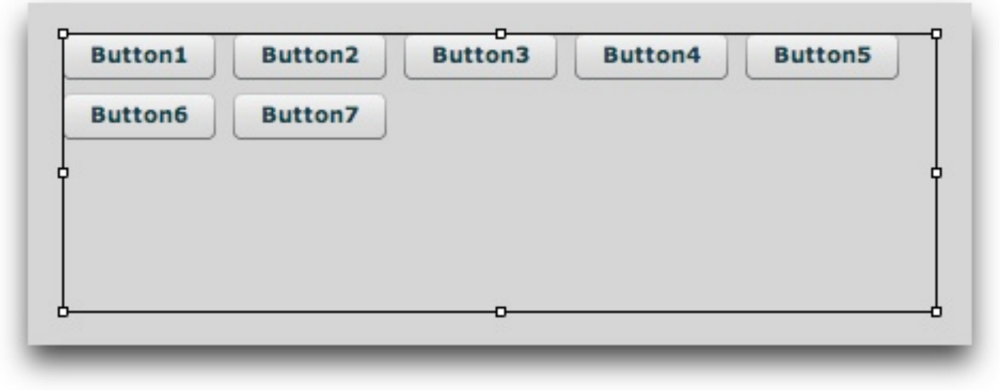
(No noteworthy specific parameters)

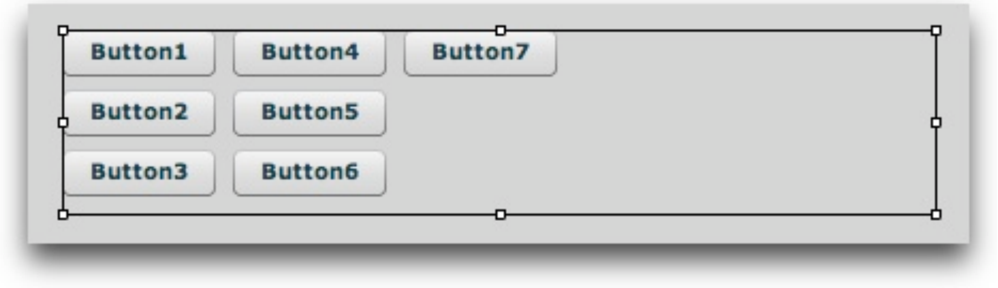
Equivalement Class File

For detailed reference on this element, see the `mx.containers.VDividedBox` class in the API Reference.

Tile

The Tile container lays out its children in a grid of commonly-sized tiles. The grid can be horizontally or vertically oriented, based on the Direction setting. If a Horizontal layout is used, children are placed in rows until that row is filled, at which point a new row is started. If a Vertical layout is used, children are placed in columns until that column is filled, at which point a new column is started. The following examples illustrate both vertical and horizontal layouts respectively. In each case, several buttons are used as child elements.





By default, the tile size is determined by the largest width and height values among all children. However, explicit Tile Width and Tile Height values can be specified to override the default calculated value. If an explicitly-defined Tile Width or Tile Height value is less than the default dimension of a child element, that element will be forced to conform to the smaller dimensions; this may result in content clipping. On the other hand, if the size of a given tile is smaller than the automatically calculated or explicitly defined tile dimensions, the Horizontal Align and Vertical Align properties specify how that child is aligned within its designated tile space.

If you explicitly specify percentage-based dimensions on a child element, the percentage is relative to the size of each tile in the Tile container. For example, specifying a width of `100%` for every child will ensure that every child expands to fill the full tile width, regardless of what its default size would have been.

Important Usage Notes

After adding the first child element, you can optionally remove the default dimensions (width / height values) from the box in the Properties Panel. In this case, the Tile container will automatically size so that a square grid is used.

If you change the Direction parameter to Horizontal (the default) to Vertical and then revert later to Horizontal, you may need to explicitly add `<Tile ... **direction="horizontal"**>` in the code for the update to be reflected in Design View.

Noteworthy Parameters

Parameter name in Properties Panel	XML Property Name	Notes
Tile Width	`tileWidth`	Can be used to explicitly specify (in pixels) the width of each tile in the container. If omitted, this value is automatically calculated to be the same as the width of the widest child element.
Tile Height	`tileHeight`	Can be used to explicitly specify (in pixels) the height of each tile in the container. If omitted, this value is automatically calculated to be the same as the height of the tallest child element.
Horizontal Gap	`horizontalGap`	Specifies, in pixels, the spacing between each column in the grid of tiles.
Vertical Gap	`verticalGap`	Specifies, in pixels, the spacing between each row in the grid of tiles.

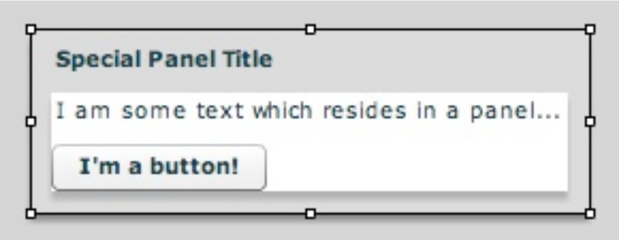
Horizontal Align	`horizontalAlign`	If a given child is narrower than than the tile which is holding it, this setting determines its horizontal alignment within that designated tile area.
Vertical Align	`verticalAlign`	If a given child is shorter than the tile which is holding it, this setting determins its vertical alignment within that designated tile area.

Equivalement Class File

For detailed reference on this element, see the `mx.containers.Tile` class in the API Reference.

Panel

The Panel container is essentially a VBox (see above) which is wrapped in a chrome box with a title. It can be used to clearly segmentize your slideshow user interface. The following example shows a panel with two example components:



Important Usage Notes

You can remove the default dimensions (width / height values) from the box in the Properties Panel. This will allow the Panel to expand as needed to accomodate its child elements. As mentioned below in the Noteworthy Parameters, the spacing between each internal edge of the Panel and the content is controlled by the `paddingLeft`, `paddingRight`, `paddingTop` and `paddingBottom` parameters, which can be specified in the XML.

Noteworthy Parameters

Parameter name in Properties Panel	XML Property Name	Notes
Title	`title`	Specifies the title text displayed on the panel.
Vertical Gap	`verticalGap`	This specifies, in pixels, the vertical spacing between the child elements.

Horizontal Align	`horizontalAlign`	When the widths of the child elements vary, this specifies the horizontal alignment of the narrower elements within the container.
Vertical Align	`verticalAlign`	If you have specified an explicit height for the VBox and the content does not fill the full height, this specifies the vertical alignment of all elements within the container.
(Not in Properties Panel)	`paddingLeft`	Specifies, in pixels, the spacing between the inner left edge of the panel and the left edge of the content.
	`paddingRight`	Specifies, in pixels, the spacing between the inner right edge of the panel and the right edge of the content.
	`paddingTop`	Specifies, in pixels, the spacing between the inner top edge of the panel and the top edge of the content.
	`paddingBottom`	Specifies, in pixels, the spacing between the inner bottom edge of the panel and the bottom edge of the content.

Equivalement Class File

For detailed reference on this element, see the `mx.containers.Panel` class in the API Reference.

HoverCanvas

The HoverCanvas container is a special type of Canvas container (detailed below). Instead of always being visible, it automatically hides itself after a period of inactivity. Another supported behavior allows for the HoverCanvas to only be visible when the mouse is in a certain target location (regardless of activity). The latter behavior is enabled by setting the Test Type property to Mouse Location (see the noteworthy parameters table below).

This HoverCanvas is commonly used to have hover-only controls which automatically hide when not in use. A common use of the Location-based behavior allows for the Caption to only be visible when the slideshow (or a particular area thereof) is hovered over. When creating control bars, it is often best to use other relative-position containers inside the Canvas (e.g. the HBox or VBox containers); this makes control-bar layout easier, as all elements are automatically placed adjacent to each other.

Two different types of hover targets can be specified through the Properties Panel: *Entire Slideshow* (default) or *Self*. If the *Entire Slideshow* option is used, the HoverCanvas will automatically re-appear whenever the user moves the mouse anywhere in the slideshow. If the *Self* option is used, the HoverCanvas will only re-appear when the user moves the mouse over the area where the container is located when visible.

Important Usage Notes

By default, a fade-in and fade-out effect are used to show and hide the HoverCanvas.   These default effects can be overridden in the Effects section of the Properties Panel.

Noteworthy Parameters

Parameter name in Properties Panel	XML Property Name	Notes
Test Type	`testType`	Specifies what triggers the HoverCanvas to hide/show itself. If Mouse Movement is used, the user must move their mouse in the specified target area to trigger visibility . On the other hand, if Mouse Location is used, the user must simply place their mouse in a target area to trigger visibility (regardless of movement).
Hover Target	`hoverTarget`	Specifies where the user must move the mouse to cause the container to automatically re-appear when hidden. If 'Entire Slideshow' is used, movement anywhere on the slideshow will trigger a re-appearance. If 'Self' is used, movement must be over the area where the container is located when visible to trigger a re-appearance.
Hide Delay	`hideDelay`	Specifies the period of inactivity required before the HoverCanvas automatically hides itself.
Initially Visible	`visible`	Specifies whether the HoverCanvas should be visible when the slideshow initially loads; usually, it is best to make the container initially visible so the user knows the controls are there.

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.HoverCanvas` class in the API Reference.

HoverSlideCanvas

The HoverCanvas container is a special type of HoverCanvas container (detailed above). It has a special built-in hide/show effect where the element hides itself by sliding out to one side of its designated area. It inherits all of the HoverCanvas's settings, plus adds two new settings detailed below.

Noteworthy Parameters

Parameter name in Properties Panel	XML Property Name	Notes
Hide Effect Side	`hideSide`	Controls the hide/show effect; specifies on which side the element hides itself in when it slides out of

		view.
Effect Speed	`hideEffectSpeed`	Specifies the duration of the hide/show effect.
Hide Delay	`hideDelay`	Specifies the period of inactivity required before the HoverCanvas automatically hides itself.
Initially Visible	`visible`	Specifies whether the HoverCanvas should be visible when the slideshow initially loads; usually, it is best to make the container initially visible so the user knows the controls are there.

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.HoverSlideCanvas` class in the API Reference.

Canvas

The Canvas element is the main absolute-position layout container. This is useful in some cases because all child elements are laid out in isolation; they can even overlap each other. X and Y positioning coordinates can be used to specify the location of each child element. Additionally, size and positioning constraints can be used to anchor the child component(s) to the edges or centers of the container, regardless of any resizing which occurs. To learn more about absolute-position containers vs. relative-position containers, see the Element Positioning and Sizing topic.

Often you will want to use a combination of absolute- and relative-position containers. This can be done by adding relative-position containers (such as the HBox or VBox) to a Canvas, then adding child components to the HBox or VBox.

Important Usage Notes

If you use X and Y positioning coordinates to position child elements, note that the coordinates are relative to the top-left corner of the container.

As with all containers, you can add a background, border and/or drop shadow to this container using the Border / Fill and Drop Shadow sections of the Properties Panel.

Noteworthy Parameters

(No noteworthy specific parameters)

Equivalement Class File

For detailed reference on this element, see the `mx.containers.Canvas` class in the API Reference.

Image Progress Loaders

Image progress loaders are included in a slideshow to show the progress of image loading, so that the user knows the status of loading operations.

Elements in this section:

- Centered Loaders Block *(a special starter element)*
- SingleLoader
- GroupLoader

Advanced Users / Developers: For more detailed developer-oriented information about every element in this section, be sure to see the API reference for each element's corresponding class.

Usage Tip: When in the wizard, hover over any property name in the Properties Panel for a detailed description of what the option controls.

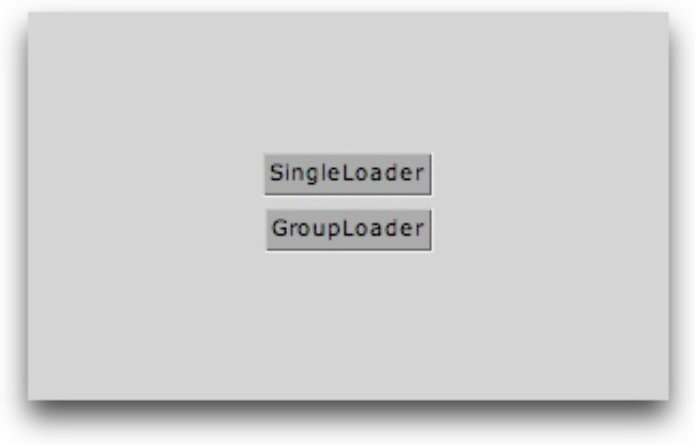
General Notes About Image Progress Loaders

You will usually want to place all loading indicators in an Overlay view (see Understanding and Managing Layout Views) so that they are always visible to the user.

Centered Loaders Block (a special starter element)

The Centered Loaders Block is a special starter element which is comprised of a VBox which holds both a centered SingleLoader and GroupLoader element. Regardless of your slideshow configuration, this element can be dropped into an overlay view to ensure that loading progress information is displayed. If you are loading images one-by-one, the SingleLoader will automatically be used; if you are using a group-preload, the GroupLoader will automatically be used.

After dropping the element into an Overlay view, you will want to make sure in the Size / Positioning section of the Properties Panel that the X and Y values are set to `0` and the Width and Height set to `100%`. This will ensure that the loaders are centered over the slideshow when they appear.



Important Usage Notes

See element description; additionally, be aware that all the individual elements which comprise a starter element can be individually edited after the starter element has been added.

Noteworthy Parameters

(For specific parameters, see the SingleLoader and GroupLoader elements below; this starter element contains both of these loading indicator elements.)

Equivalement Class File

This is a compound starter element made up of other elements; it has no individual associated class file.

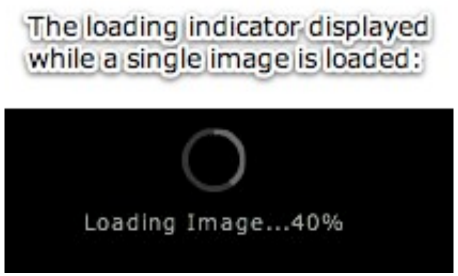
The Layout XML for the starter element is as follows:

```
<VBox x="0" y="0" width="100%" height="100%" horizontalAlign="center" verticalAlign="middle">
     <SingleLoader />
     <GroupLoader />
</VBox>
```

SingleLoader

The SingleLoader element displays the progress of all individual-image loading operations. When a single image begins to load, it automatically appears;

when the load completes, it automatically hides. The appearance of the loader can be customized; see the Noteworthy Parameters below for more information. Here is an example of the element in use within a slideshow:


The loading indicator displayed while a single image is loaded:

To ensure that the progress indicator is always visible when loading takes place, it is recommended that you place it in an Overlay layout view.

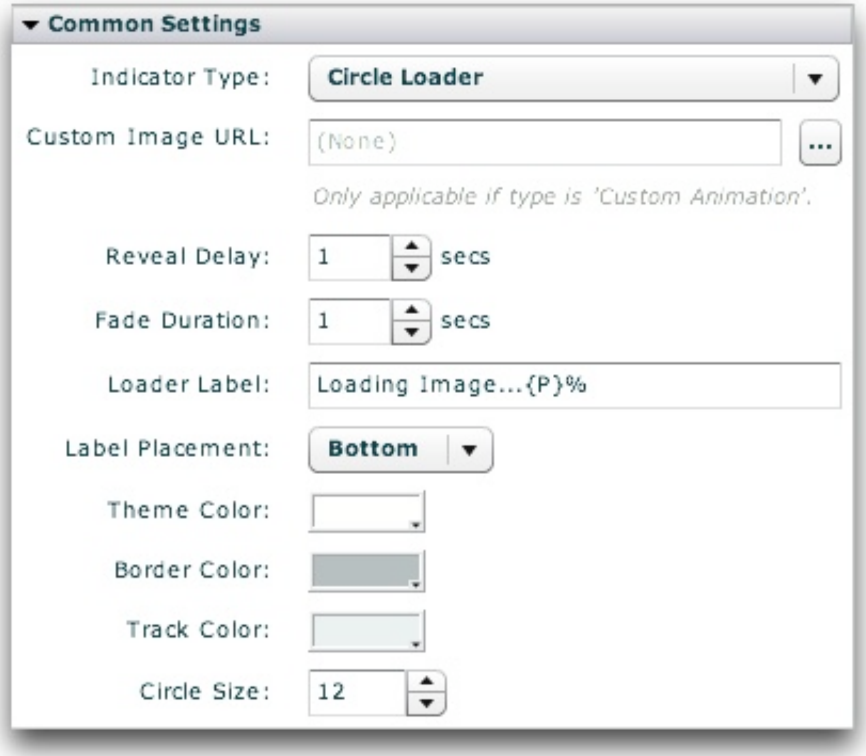
Important Usage Notes

It is usually best to include both SingleLoader and GroupLoader progress indicator elements in your slideshow. Then, if at some point the loading method is changed, loading progress will still be shown. You can save time and use the Centered Loaders Block starter element, which includes both loaders in a centered container. See above for more information about this starter element.

In addition to the parameters listed in the Noteworth Parameters table below, there are many other settings which can be used to customize the appearance of the loader. Some of these are included directly in the Properties Panel; others are described in the `SingleLoader` class file referenced below.

Noteworthy Parameters

The Common Settings section of the Properties Panel contains many noteworthy settings; they are described in the table below:



Common Settings

Indicator Type: Circle Loader

Custom Image URL: (None) ...

Only applicable if type is 'Custom Animation'.

Reveal Delay: 1 secs

Fade Duration: 1 secs

Loader Label: Loading Image...{P}%

Label Placement: Bottom

Theme Color:

Border Color:

Track Color:

Circle Size: 12

Parameter name in Properties Panel	XML Property Name	Notes
Indicator Type	`type`	This specifies the type of indicator shown. Supported options include: ▪ **Circle Loader** - (XML attribute value of `ProgressCircle`) - Displays a circular loader similar to the one shown above in the element example. The circle/ring fills as the loading takes place. ▪ **Progress Bar** - (XML attribute value of `ProgressBar`) - Displays a standard progress bar. The appearance of the bar is mostly controlled by the color settings in the Common Settings section, especially the Theme Color. ▪ **Progress Line** - (XML attribute value of `ProgressLine`) - Displays a progress bar which is only one pixel thick; more subtle than other loaders. The appearance of the line is mostly controlled by the color settings in the Common Settings section, especially the Theme Color.

		- **Progress Zip Line** - (XML attribute value of `ProgressZipLine`) - Much like the Progress Line, except that there is an animation which "uncompresses" the loader's width, lets it fill, then "compresses" it again down to nothing to hide it. - **Custom Animation** - (XML attribute value of `ProgressImage`) - Allows you to use a custom loader image, as specified in the Custom Image URL field.
Custom Image URL	`customImageURL`	If *Custom Animation* is selected as the Indicator Type, this value specifies the URL to the custom loading indicator image/animation which will be displayed. The file can be a JPEG, PNG, SWF, or GIF. The URL must be relative to the HTML page holding the slideshow; use the browse button ('...') to automatically generate an appropriate URL. If you attempt to load content from another domain, you may be affected by Flash Player cross-domain loading security restrictions.
Reveal Delay	`showDelay`	By default, the loader doesn't immediately appear when a load starts. Instead, there is a short delay (1 second by default) which ensures that the loader isn't constantly flashing in and out for short loads. In other words, the loader will only appear if the load takes more than the time specified by Reveal Delay. A shorter value may make the loader appear more often, whereas a longer value may confuse users because they won't know what's happening.
Fade Duration	`fadeTime`	When the loader appears and disappears, a fade effect is automatically applied. This parameter specifies the duration of thes effect.
Loader Label	`baseLabel`	Specifies the label displayed on the loader. To disable the label, enter an empty string or a single space. If `{P}` is used within the label, it will be automatically replaced with the percentage of loading which has completed. For example, the default label of "Loading Image...{P}%" displays "Loading Image...50%" when a load operation is halfway complete.
Theme Color	`themeColor`	Specifies the base color used for the loading indicator.
Circle Size	`innerRadius`	If using a Circle Loader as the Indicator Type, specifies the size of the loading indicator.

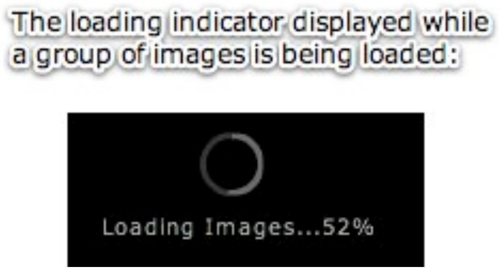
Important Note: *For more information about all other parameters, including those not listed above and those not accessible through the Properties Panel, see the class reference mentioned below.*

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.SingleLoader` class in the API Reference.

GroupLoader

The GroupLoader element displays the progress of all image-loading operations when more than one image is being loaded. When a group of images begins to load, it automatically appears; when the load completes, it automatically hides. The appearance of the loader can be customized; see the Noteworthy Parameters below for more information. Here is an example of the element in use within a slideshow:


The loading indicator displayed while a group of images is being loaded:

To ensure that the progress indicator is always visible when loading takes place, it is recommended that you place it in an Overlay layout view.

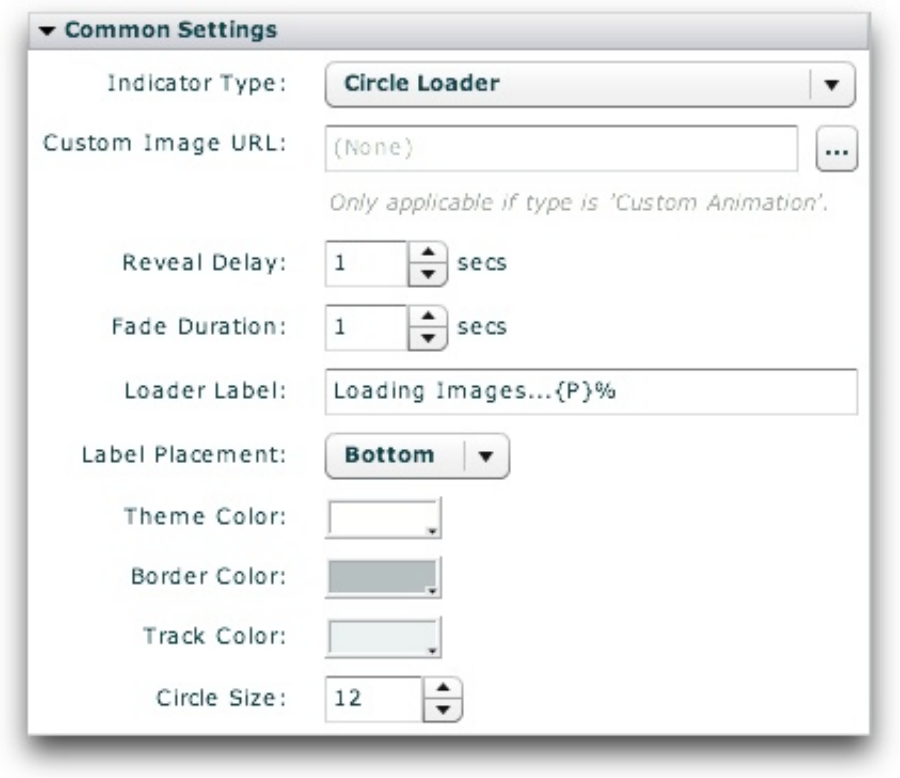
Important Usage Notes

It is usually best to include both SingleLoader and GroupLoader progress indicator elements in your slideshow. Then, if at some point the loading method is changed, loading progress will still be shown. You can save time and use the Centered Loaders Block starter element, which includes both loaders in a centered container. See above for more information about this starter element.

In addition to the parameters listed in the Noteworth Parameters table below, there are many other settings which can be used to customize the appearance of the loader. Some of these are included directly in the Properties Panel; others are described in the `GroupLoader` class file referenced below.

Noteworthy Parameters

The Common Settings section of the Properties Panel contains many noteworthy settings; they are described in the table below:



Common Settings

Indicator Type: Circle Loader

Custom Image URL: (None)

Only applicable if type is 'Custom Animation'.

Reveal Delay: 1 secs

Fade Duration: 1 secs

Loader Label: Loading Images...{P}%

Label Placement: Bottom

Theme Color:

Border Color:

Track Color:

Circle Size: 12

Parameter name in Properties Panel	**XML Property Name**	**Notes**
Indicator Type	`type`	This specifies the type of indicator shown. Supported options include: ▪ **Circle Loader** - (XML attribute value of `ProgressCircle`) - Displays a circular loader similar to the one shown above in the element example. The circle/ring fills as the loading takes place. ▪ **Progress Bar** - (XML attribute value of `ProgressBar`) - Displays a standard progress bar. The appearance of the bar is mostly controlled by the color settings in the Common Settings section, especially the Theme Color. ▪ **Progress Line** - (XML attribute value of `ProgressLine`) - Displays a progress bar which is only one pixel thick; more subtle than other loaders. The appearance of the line is mostly controlled by the color settings in the Common Settings section, especially the Theme Color.

		- **Progress Zip Line** - (XML attribute value of `ProgressZipLine`) - Much like the Progress Line, except that there is an animation which "uncompresses" the loader's width, lets it fill, then "compresses" it again down to nothing to hide it. - **Custom Animation** - (XML attribute value of `ProgressImage`) - Allows you to use a custom loader image, as specified in the Custom Image URL field.
Custom Image URL	`customImageURL`	If *Custom Animation* is selected as the Indicator Type, this value specifies the URL to the custom loading indicator image/animation which will be displayed. The file can be a JPEG, PNG, SWF, or GIF. The URL must be relative to the HTML page holding the slideshow; use the browse button ('...') to automatically generate an appropriate URL. If you attempt to load content from another domain, you may be affected by Flash Player cross-domain loading security restrictions.
Reveal Delay	`showDelay`	By default, the loader doesn't immediately appear when a load starts. Instead, there is a short delay (1 second by default) which ensures that the loader isn't constantly flashing in and out for short loads. In other words, the loader will only appear if the load takes more than the time specified by Reveal Delay. A shorter value may make the loader appear more often, whereas a longer value may confuse users because they won't know what's happening.
Fade Duration	`fadeTime`	When the loader appears and disappears, a fade effect is automatically applied. This parameter specifies the duration of thes effect.
Loader Label	`baseLabel`	Specifies the label displayed on the loader. To disable the label, enter an empty string or a single space. If `{P}` is used within the label, it will be automatically replaced with the percentage of loading which has completed. For example, the default label of "Loading Image...{P}%" displays "Loading Image...50%" when a load operation is halfway complete.
Theme Color	`themeColor`	Specifies the base color used for the loading indicator.
Circle Size	`innerRadius`	If using a Circle Loader as the Indicator Type, specifies the size of the loading indicator.

Important Note: *For more information about all other parameters, including those not listed above and those not accessible through the Properties Panel, see the class reference mentioned below.*

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.GroupLoader` class in the API Reference.

Slideshow Image Displays

Image display elements are perhaps the most important class of elements which are added to a slideshow. These elements display all of the slides which you have added in the wizard or loaded from external data sources.

To use a physical analogy, your slideshow could be compared to a movie room. You might have some miscellaneous "elements," such as couches and chairs, which contribute to the overall atmosphere and make the experience more "usable." However, the centerpiece is the television screen on which the movies appear; this is comparable to the image display in your slideshow.

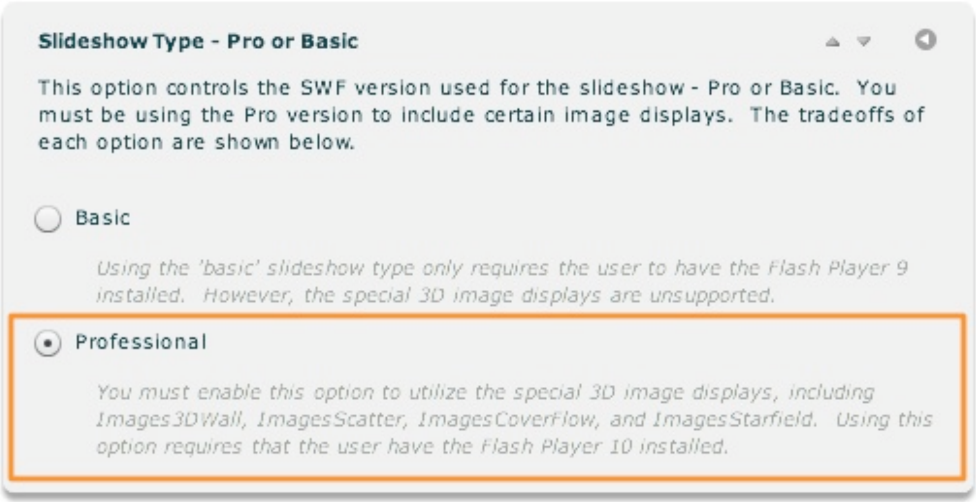
There are a number of different image display elements available. The most commonly used display is the ImagesStandard element; this provides a traditional slideshow visual experience. However, there is also the option of using more impressive and interactive 3D image displays. Most of these alternative displays are only available in the Pro version of the software.

Elements in this section:

- ImagesStandard (most commonly used)
- ImagesScrollList
- ImagesStarfield* (Pro version only)
- Images3DWall* (Pro version only)
- ImagesCoverFlow* (Pro version only)
- ImagesScatter* (Pro version only)

***Important note about using the Pro-version-only image display elements:**

When using the ImagesStarfield, Images3DWall, ImagesCoverFlow, or ImagesScatter image display elements, you must set the Slideshow Type to **Professional** in the "Slideshow Type - Pro or Basic" pod in the Global Configuration and Playback Settings section:



Slideshow Type - Pro or Basic

This option controls the SWF version used for the slideshow - Pro or Basic. You must be using the Pro version to include certain image displays. The tradeoffs of each option are shown below.

○ Basic

Using the 'basic' slideshow type only requires the user to have the Flash Player 9 installed. However, the special 3D image displays are unsupported.

● Professional

You must enable this option to utilize the special 3D image displays, including Images3DWall, ImagesScatter, ImagesCoverFlow, and ImagesStarfield. Using this option requires that the user have the Flash Player 10 installed.

If you forget to change this setting, an "unable to render" message will be displayed when you attempt to view the slideshow.

Advanced Users / Developers: For more detailed developer-oriented information about every element in this section, be sure to see the API reference for each element's corresponding class.

ImagesStandard

The ImagesStandard element is the most commonly used image display. All "image transition" settings throughout the wizard apply to slideshows which use this element. It provides a traditional slide display, with one slide visible at a time and a variety of transitions available to go from one slide to another. To learn about the available image transitions, see the Setting the Image Transition topic within Using the 'Global Configuration and Playback Settings' Section. In addition to setting the global default transition, you can also specify gallery- or image-specific transitions.

Here is an example of the ImagesStandard element in use within a slideshow:

The standard image display which displays all the slides defined in the slideshow's image galleries.

Important Usage Notes

The actual size at which slide images appears depends on the size of the image display element (ImagesStandard in this case), as well as the Image Scaling Mode specified in the main Slide Appearance tab.

Noteworthy Parameters

(None)

Equivalement Class File

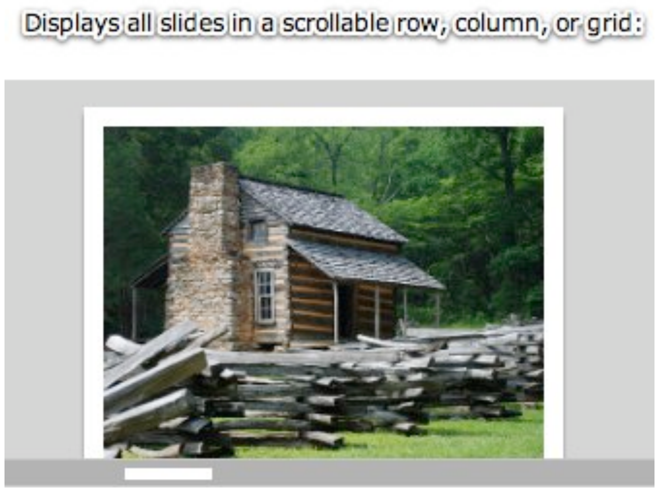
For detailed reference on this element, see the `com.dwuser.ss4.ui.ImagesStandard` class in the API Reference.

ImagesScrollList

The ImagesScrollList elements is the second image display element available in the Basic version of the software. It is designed for situations when you want all slides to appear directly adjacent to each other. This allows the user to scroll through all slides at once, either manually or via hover-based movement. The images can be arranged in a scrollable row, column, or square grid.

Because all slides are displayed at once, this image display is best used with smaller galleries (e.g. 50 images or less). While you can use it with larger galleries, performance degradation may occur, especially on older systems.

Here is an example of the ImagesScrollList element in use within a slideshow:



Displays all slides in a scrollable row, column, or grid:

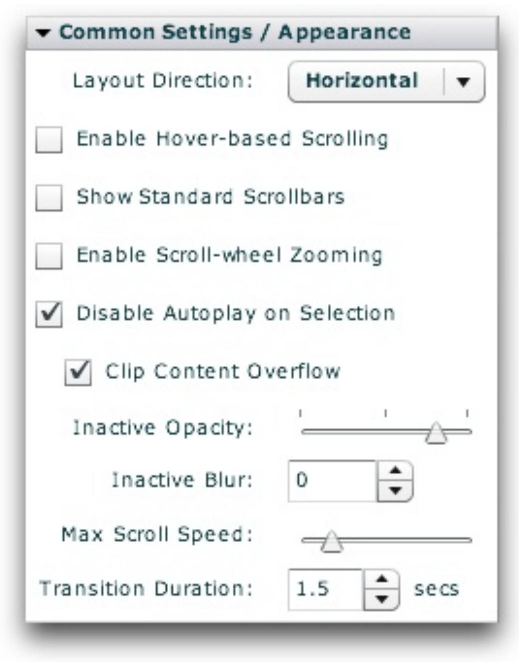
Important Usage Notes

The actual size at which slide images appears depends on the size of the image display element, as well as the Image Scaling Mode specified in the main Slide Appearance tab.

If you want any content which overflows the defined element bounds to still be visible, disable the Clip Content Overflow setting (see the parameters table below for more information).

Noteworthy Parameters

The Common Settings section of the Properties Panel contains many noteworthy settings; they are described in the table below:



Common Settings / Appearance

Layout Direction: Horizontal

□ Enable Hover-based Scrolling

□ Show Standard Scrollbars

□ Enable Scroll-wheel Zooming

☑ Disable Autoplay on Selection

☑ Clip Content Overflow

Inactive Opacity:

Inactive Blur: 0

Max Scroll Speed:

Transition Duration: 1.5 secs

Parameter name in Properties Panel	XML Property Name	Notes
Layout Direction	`layoutDirection`	Specifies how the slides should be arranged; supported values are Horizontal (the default; creates one long row of all slides), Vertical (creates one long column of all slides), and Grid (creates a large square grid of all slides).
Enable Hover-based Scrolling	`useMouseScroll`	Specifies if the user can scroll the entire layout grid by simply moving his/her mouse away from the center of the display. The downside to this scrolling method is that the user must actually click a slide to activate it. When this scrolling method is enabled, you generally will want to set Inactive Opacity to `1.0` and Inactive Blur to `0`. If this setting is enabled, you can control the maximum scrolling speed through the Max Scroll Speed setting.
Show Standard Scrollbars	`useScrollbars`	Specifies if standard scrollbars should be shown, allowing the user to navigate the grid with scrollbars.
Enable Scroll-	`useScrollWheel`	Specifies if the user can zoom in and out of the grid by using the mouse scroll wheel in order to gain a

wheel Zooming		wider or narrower perspective.
Disable Autoplay on Selection	`disableAutoplayOnClick`	Specifies if autoplay should automatically be disabled/paused when the user manually selects a slide by clicking it.
Clip Content Overflow	`clipContent`	Specifies if any content which exceeds the specified bounds for the image display should be clipped (hidden) or not.
Inactive Opacity	`inactiveOpacity`	Specifies the opacity (alpha) of all slides which are not currently focused upon. This setting allows you to put all focus on the main slide by reducing the visual distraction of any other visible slides. Valid values range from 0.0 (fully transparent) to 1.0 (fully opaque).
Inactive Blur	`inactiveBlur`	Specifies the strength of the blur applied to all slides which are not currently focused upon. This setting allows you to put all focus on the main slide by reducing the visual distraction of any other visible slides. Leaving this value at its default, 0, disables the blur.
Max Scroll Speed	`maxScrollSpeed`	Specifies the maximum auto-scrolling speed when using mouse-based scrolling as enabled through the 'Enable Hover-based Scrolling' setting. This is the maximum distance the layout grid can move per update. The default value is 20 (pixels).
Transition Duration	`duration`	Specifies the duration of the automated transition between each slide (e.g. when autoplay is enabled).

Important Note: *For more information about all other parameters, including those not listed above and those not accessible through the Properties Panel, see the class reference mentioned below.*

Equivalement Class File

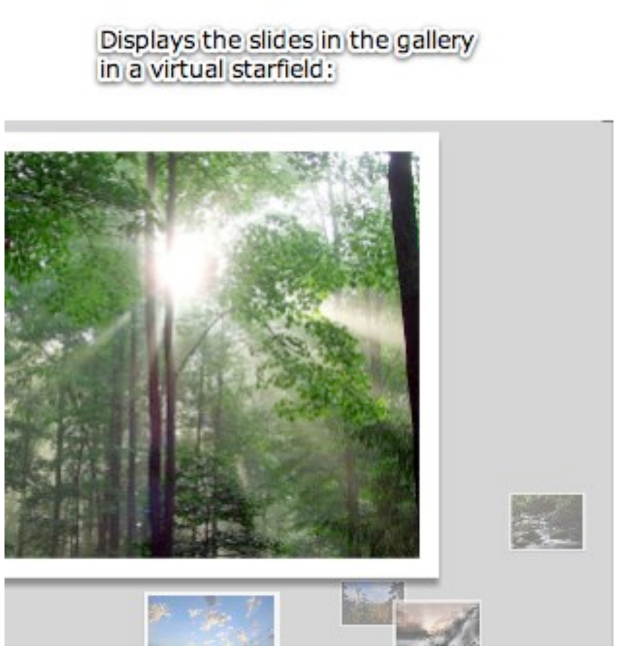
For detailed reference on this element, see the `com.dwuser.ss4.ui.ImagesScrollList` class in the API Reference.

ImagesStarfield

The ImagesStarfield element displays all slides within a virtual 3D starfield, with a flight-like effect to transition between slides.   It is one of the Pro-version-only image display elements.   **Important:** The Professional value must be selected for Slideshow Type when using it; see the note at the top of this topic for more information.

Because all slides are displayed at once, this image display is best used with smaller galleries (e.g. 50 images or less).   While you can use it with larger galleries, performance degradation may occur, especially on older systems.

Here is an example of the ImagesStarfield element in use within a slideshow:



Displays the slides in the gallery
in a virtual starfield:

Important Usage Notes

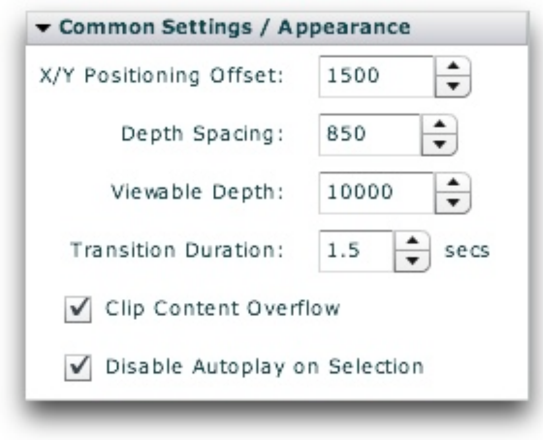
The actual size at which slide images appears depends on the size of the image display element, as well as the Image Scaling Mode specified in the main Slide Appearance tab.

If you want any content which overflows the defined element bounds to still be visible, disable the Clip Content Overflow setting (see the parameters table below for more information).

Again, the Professional value must be selected for Slideshow Type when using this element; see the note at the top of this topic for more information.

Noteworthy Parameters

The Common Settings section of the Properties Panel contains many noteworthy settings; they are described in the table below:



Common Settings / Appearance

X/Y Positioning Offset: 1500

Depth Spacing: 850

Viewable Depth: 10000

Transition Duration: 1.5 secs

☑ Clip Content Overflow

☑ Disable Autoplay on Selection

Parameter name in Properties Panel	XML Property Name	Notes
X/Y Positioning Offset	`positioningOffset`	Specifies, in pixels, the maximum x and y offset of each slide from the center (the focal point). Increasing this value makes the slides appear more randomly distributed in the starfield. The default value is 1500.
Depth Spacing	`depthSpacing`	Specifies the z-depth spacing between each image in the starfield (in pixels). Increasing this value makes the slides seem farther apart and the starfield more vast. The default value is 850.
Viewable Depth	`viewableDepth`	Specifies the depth of the field of vision (in pixels). This controls the z distance behind the current slide which is visible, before it fades entirely away into the distance. The default value is 10000.
Transition Duration	`duration`	Specifies the duration of the automated transition between each slide (e.g. when autoplay is enabled).
Clip Content Overflow	`clipContent`	Specifies if any content which exceeds the specified bounds for the image display should be clipped (hidden) or not.
Disable Autoplay on Selection	`disableAutoplayOnClick`	Specifies if autoplay should automatically be disabled/paused when the user manually selects a slide by clicking it.

Important Note: *For more information about all other parameters, including those not listed above and those not accessible through the Properties Panel, see the class reference mentioned below.*

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.ImagesStarfield` class in the API Reference.

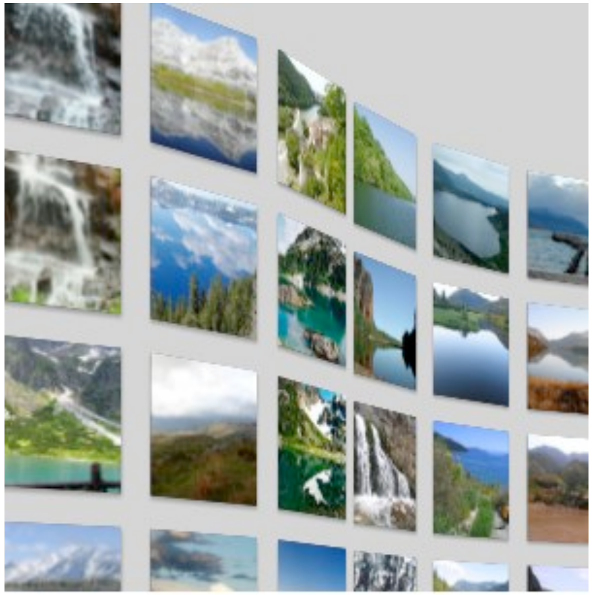
Images3DWall

The Images3DWall element is a 3D image display where all images are laid out on a three dimensional "wall." The user can zoom in and out, drag the wall around, and focus in on single images. The wall can be curved, shaped like a wave, or even made into a circular carousel. It is one of the Pro-version-only image display elements. **Important:** The Professional value must be selected for Slideshow Type when using it; see the note at the top of this topic for more information.

The Images3DWall creates a large grid of all the current gallery's thumbnails and arranges them on the virtual 3D wall. When a thumb is clicked on, the camera flys in to focus on the image, and the full-quality image is loaded and displayed. If any video or audio have been added to the slide, they are played at this point. The user can click outside the slide's boundaries to return to the main overview, or use the scroll wheel to zoom back out.

This image display should be used in conjunction with the Preload Thumbs preloading mode setting (as set in the Global Configuration and Playback Settings section). This will ensure that all thumbs for the gallery are initially loaded before the user begins exploring the 3D wall.

Here is an example of the Images3DWall element in use within a slideshow:



Displays all slides in a virtual 3D environment, with the slides spread across a 'wall.' A carousel shape or totally flat wall can also be used.

Important Usage Notes

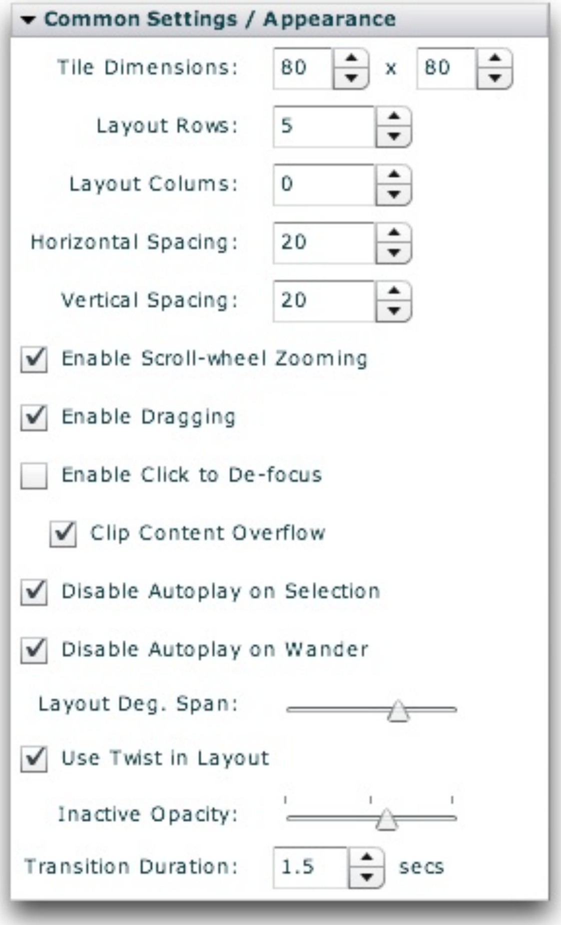
The actual size at which slide images appears depends on the size of the image display element, as well as the Image Scaling Mode specified in the main Slide Appearance tab.

If you want any content which overflows the defined element bounds to still be visible, disable the Clip Content Overflow setting (see the parameters table below for more information).

Again, the Professional value must be selected for Slideshow Type when using this element; see the note at the top of this topic for more information.

Noteworthy Parameters

The Common Settings section of the Properties Panel contains many noteworthy settings; they are described in the table below:



Common Settings / Appearance

Tile Dimensions: 80 x 80

Layout Rows: 5

Layout Colums: 0

Horizontal Spacing: 20

Vertical Spacing: 20

☑ Enable Scroll-wheel Zooming

☑ Enable Dragging

☐ Enable Click to De-focus

☑ Clip Content Overflow

☑ Disable Autoplay on Selection

☑ Disable Autoplay on Wander

Layout Deg. Span:

☑ Use Twist in Layout

Inactive Opacity:

Transition Duration: 1.5 secs

Parameter name in Properties Panel	XML Property Name	Notes
Tile Dimensions	`tileWidth`	Specifies, in pixels, the width of each thumbnail tile on the 3D wall.
	`tileHeight`	Specifies, in pixels, the height of each thumbnail tile on the 3D wall.
Layout Rows	`layoutRows`	Specifies the number of rows of thumbs on the 3D wall's grid layout. This value is used to create a layout which is always the same height, but of variable width. It can be overridden by setting the Layout Columns property. The default value is 5.
Layout Columns	`layoutColumns`	Specifies the number of columns of thumbs on the 3D wall's grid layout. If set to a value greater than 0, this property overrides the Layout Rows property. Specifying a value on this property creates a layout which is always the same width, but of variable height.
Horizontal Spacing	`horizontalSpacing`	Specifies, in pixels, the horizontal spacing between each thumbnail tile on the 3D wall. The default value is 20.
Vertical Spacing	`verticalSpacing`	Specifies, in pixels, the vertical spacing between each thumbnail tile on the 3D wall. The default value is 20.
Enable Scroll-wheel Zooming	`useScrollWheel`	Specifies if scroll-wheel interaction is enabled. If enabled, the user can zoom in and out of the 3D wall using the mouse's scroll wheel.
Enable Dragging	`useDrag`	Specifies if the user can drag the wall around to change its relative position. If enabled, the user can drag the wall in order to see a different portion.
Enable Click to De-focus	`useClick`	Specifies if the user can click on an empty area to zoom back out from a slide, returning to the full-wall display. Keep in mind that if the user clicks on a different slide, they will zoom to *that* slide, not back to the full-wall display.
Clip Content Overflow	`clipContent`	Specifies if any content which exceeds the specified bounds for the image display should be clipped (hidden) or not.
Disable Autoplay on Selection	`disableAutoplayOnClick`	Specifies if autoplay should automatically be disabled/paused when the user manually selects a slide by clicking it.
Disable Autoplay on Wander	`disableAutoplayOnWander`	Specifies if autoplay should automatically be disabled/paused when the user wanders away from a slide by clicking in an empty area (to return to the full-wall display), dragging the wall to change its position, or using the scroll-wheel to zoom in or out.
Layout Degree Span	`layoutDegreeSpan`	Specifies the number of degrees spanned by the curve on the 3D wall's grid layout. Valid values range from `-360` to `360`; using negative values reverses the direction of the curve. Using a value of `360` or `-360` makes a full circle, giving a carousel-like appearance. Using a value of `0` disables any curvature, making a flat wall.

		If you wish to make the wall curve back the opposite direction at the middle (giving a wave-like appearance), enable the Use Twist in Layout setting. The default value is `120`.
Use Twist in Layout	`layoutUseTwist`	Specifies if the 3D wall's layout should include a "twist", or a reversal of curvature direction at the middle. See the Layout Degree Span setting for more information. When enabled, the wall curves one direction then curves back the other, giving a wave-like appearance.
Inactive Opacity	`shadingOpacity`	Specifies the opacity (alpha) of the non-active thumbnails when a certain slide is focused on. Valid values range from `0.0` (fully transparent) to `1.0` (fully opaque); the default value is `0.6`.
Transition Duration	`duration`	Specifies the duration of the automated transition between each slide (e.g. when autoplay is enabled).

Important Note: *For more information about all other parameters, including those not listed above and those not accessible through the Properties Panel, see the class reference mentioned below.*

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.Images3DWall` class in the API Reference.

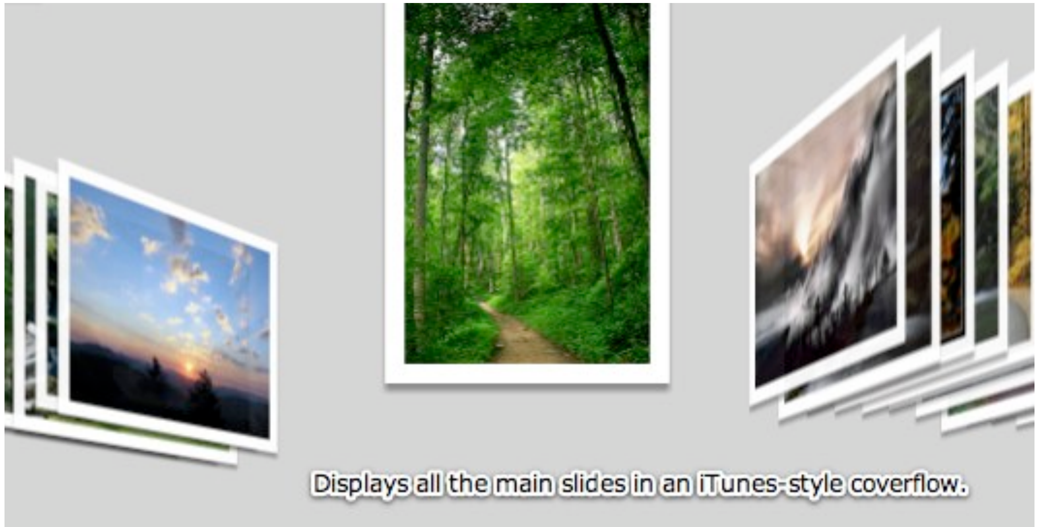
ImagesCoverFlow

The ImagesCoverFlow element creates a coverflow display of all slides in the gallery, similar to an iPod® or iTunes® coverflow. If you're not familiar with what coverflows are, see the pictorial example shown below.   It is one of the Pro-version-only image display elements.   **Important:** The Professional value must be selected for Slideshow Type when using it; see the note at the top of this topic for more information.

Because all slides are displayed at once, this image display is best used with smaller galleries (e.g. 50 images or less).   While you can use it with larger galleries, performance degradation may occur, especially on older systems.

This image display can be used in conjunction with any preloading mode setting (as set in the Global Configuration and Playback Settings section), but generally either Smart or Preload All should be used. You can also use either Preload Thumbs or 3D.   For the very best performance, you should use low-quality version of the images for the main image source, and the regular high-quality images as the High Quality Image source. This will allow the low-quality images to quickly load, and when a user clicks on a specific slide in the coverflow, a higher-quality version will then be loaded and displayed.   To learn more about specifying a separate high-quality image, see the Image Properties section.

Here is an example of the ImagesCoverFlow element in use within a slideshow:



Important Usage Notes

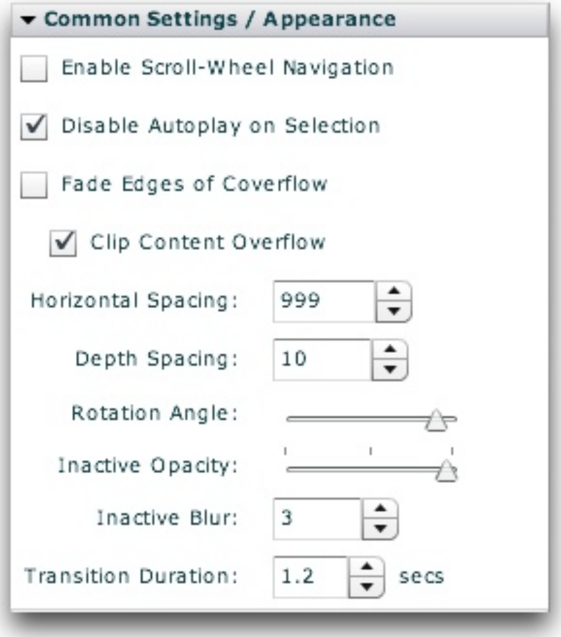
The actual size at which slide images appears depends on the size of the image display element, as well as the Image Scaling Mode specified in the main Slide Appearance tab.

If you want any content which overflows the defined element bounds to still be visible, disable the Clip Content Overflow setting (see the parameters table below for more information).

Again, the Professional value must be selected for Slideshow Type when using this element; see the note at the top of this topic for more information.

Noteworthy Parameters

The Common Settings section of the Properties Panel contains many noteworthy settings; they are described in the table below:



Common Settings / Appearance

□ Enable Scroll-Wheel Navigation

☑ Disable Autoplay on Selection

□ Fade Edges of Coverflow

☑ Clip Content Overflow

Horizontal Spacing: 999

Depth Spacing: 10

Rotation Angle:

Inactive Opacity:

Inactive Blur: 3

Transition Duration: 1.2 secs

Parameter name in Properties Panel	XML Property Name	Notes
Enable Scroll-Wheel Navigation	`useScrollWheel`	If enabled, the user can move from slide to slide in the coverflow by using the scroll wheel on the mouse.
Disable Autoplay on Selection	`disableAutoplayOnClick`	Specifies if autoplay should automatically be disabled/paused when the user manually selects a slide by clicking it.
Fade Edges of Coverflow	`fadeEdges`	If enabled, the slides closer to the edges (away from the current slide) will fade away.
Clip Content Overflow	`clipContent`	Specifies if any content which exceeds the specified bounds for the image display should be clipped (hidden) or not.
Horizontal Spacing	`horizontalSpacing`	Specifies the horizontal spacing, in pixels, between each slide in the coverflow. By default, this value is calculated to equal (Component Width) / 5. To use this default value, don't modify the value or set it to 999 (which triggers the calculated value).
Depth Spacing	`verticalSpacing`	Specifies the depth spacing between each slide in the coverflow; this setting allows you to make the out-of-focus slides fade into the distance. The setting is used in determining the z-position of each image.

Rotation Angle	`rotationAngle`	Specifies the angle at which each inactive slide in the coverflow is rotated.
Inactive Opacity	`inactiveOpacity`	Specifies the opacity (alpha) of the slides which are not currently focused upon.
Inactive Blur	`inactiveBlur`	Specifies the strength (in pixels) of the blur applied to slides which are not currently focused upon. To use no blur, set to 0.
Transition Duration	`duration`	Specifies the duration of the transition effect which occurs between slides.

Important Note: *For more information about all other parameters, including those not listed above and those not accessible through the Properties Panel, see the class reference mentioned below.*

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.ImagesCoverFlow` class in the API Reference.

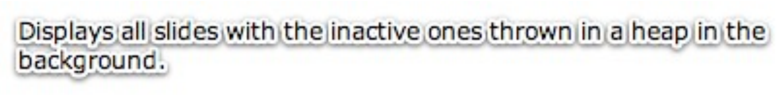
ImagesScatter

The ImagesScatter element scatters all slides in a gallery randomly across a virtual table. The slides are all positioned in the general area of the z-depth specified by the Inactive Z-Depth parameter, making them appear smaller. When one of the slides is clicked, it moves up into focus, displaying at its normal size. It is one of the Pro-version-only image display elements. **Important:** The Professional value must be selected for Slideshow Type when using it; see the note at the top of this topic for more information.

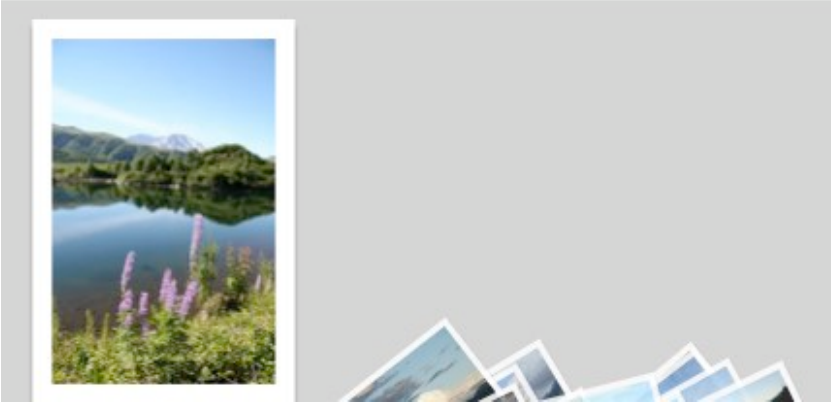
Because all slides are displayed at once, this image display is best used with smaller galleries (e.g. 50 images or less). While you can use it with larger galleries, performance degradation may occur, especially on older systems.

This image display can be used in conjunction with any preloading mode setting (as set in the Global Configuration and Playback Settings section), but generally either Smart or Preload All should be used. You can also use either Preload Thumbs or 3D. For the very best performance, you should use low-quality version of the images for the main image source, and the regular high-quality images as the High Quality Image source. This will allow the low-quality images to quickly load, and when a user clicks on a specific slide in the scatter, a higher-quality version will then be loaded and displayed. To learn more about specifying a separate high-quality image, see the Image Properties section.

Here is an example of the ImagesScatter element in use within a slideshow:



Displays all slides with the inactive ones thrown in a heap in the background.



Important Usage Notes

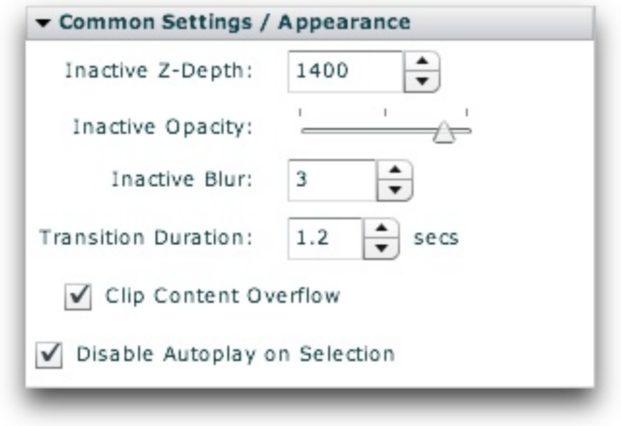
The actual size at which slide images appears depends on the size of the image display element, as well as the Image Scaling Mode specified in the main Slide Appearance tab.

If you want any content which overflows the defined element bounds to still be visible, disable the Clip Content Overflow setting (see the parameters table below for more information).

Again, the Professional value must be selected for Slideshow Type when using this element; see the note at the top of this topic for more information.

Noteworthy Parameters

The Common Settings section of the Properties Panel contains many noteworthy settings; they are described in the table below:



Common Settings / Appearance

Inactive Z-Depth: 1400

Inactive Opacity:

Inactive Blur: 3

Transition Duration: 1.2 secs

☑ Clip Content Overflow

☑ Disable Autoplay on Selection

Parameter name in Properties Panel	XML Property Name	Notes
Inactive Z-Depth	`inactiveZ`	Specifies the z-depth of the slides which are inactive. Use a larger value to make them seem farther away.
Inactive Opacity	`inactiveOpacity`	Specifies the opacity (alpha) of the slides which are not currently focused upon.
Inactive Blur	`inactiveBlur`	Specifies the strength (in pixels) of the blur applied to slides which are not currently focused upon. To use no blur, set to 0.
Transition Duration	`duration`	Specifies the duration of the transition effect which occurs between slides.
Clip Content Overflow	`horizontalSpacing`	If enabled, all content which doesn't fit in the defined dimensions will be clipped. If disabled, this overflow content will still be visible.
Disable Autoplay on Selection	`disableAutoplayOnClick`	If enabled, autoplay will automatically be disabled when the user clicks/selects a slide.

Important Note: *For more information about all other parameters, including those not listed above and those not accessible through the Properties Panel, see the class reference mentioned below.*

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.ImagesScatter` class in the API Reference.

Main Slideshow Elements

The Main Slideshow Elements are a class of essential slideshow components which are used when building layouts. The common thread is that this class of elements provides a means to **organize** and **navigate** the gallery and image data you have added to your slideshow. For example -- gallery selection elements (needed if more than one gallery exists), caption and dynamic text (for displaying the caption, title, etc. associated with a slide), and standard navigation modules such as Thumbnails.

Elements in this section:

- Pre-Built Hover-only Control Bar - NumberBar
- Pre-Built Hover-only Control Bar - Thumbnails
- Pre-Built Control Bar - NumberBar
- Pre-Built Control Bar - Thumbnails
- ImageTree
- Thumbnails
- NumberBar
- GallerySelector
- GalleryBar
- DynamicScrollableText
- DynamicLinkButton
- Caption
- DynamicText

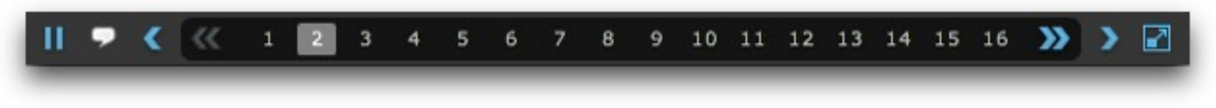
Advanced Users / Developers: For more detailed developer-oriented information about every element in this section, be sure to see the API reference for each element's corresponding class.

Tip: All of the elements within this section can be placed in a HoverCanvas element, which makes them only visible when the user actively moves the mouse over the slideshow. See the HoverCanvas layout container for more.

Pre-Built Hover-only Control Bar - NumberBar

This special starter element is a horizontal control bar which contains a NumberBar; because it is inside a HoverCanvas, **it is only visible when the user moves their mouse**. It is designed to go at the bottom of the slideshow by default, but may be moved anywhere. The default buttons can be customized as needed. Buttons can also be added or removed.

The following image shows one of these control bars (with default configuration) in use within a slideshow:



Note: All the individual elements which comprise a starter element can be individually edited after the starter element has been added.

Equivalement Class File

This is a compound starter element made up of other elements; it has no individual associated class file.

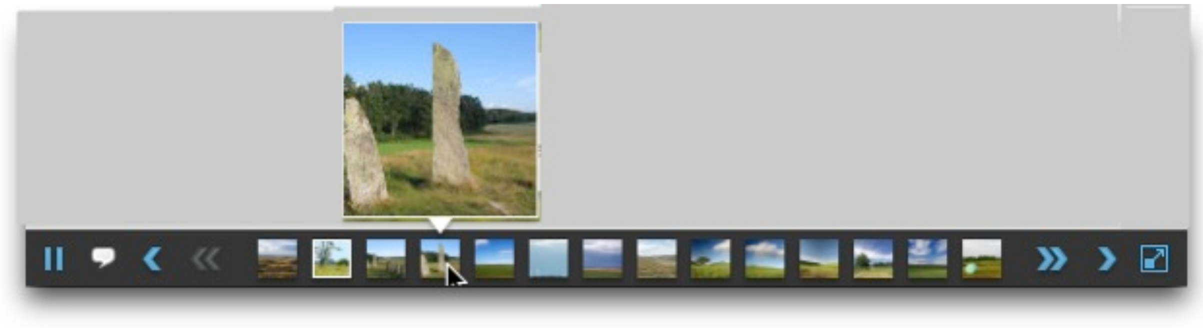
The Layout XML for the starter element is as follows:

```
<HoverCanvas left="0" right="0" bottom="0">
    <HBox left="0" right="0" bottom="0" paddingLeft="10" paddingRight="10" paddingTop="3"
paddingBottom="3" horizontalGap="10" backgroundColor="#000000" backgroundAlpha="0.8"
verticalAlign="middle">
        <PlayPauseButton/>
        <CaptionToggleButton/>
        <PreviousButton/>
        <NumberBar width="100%"/>
        <NextButton/>
        <FullScreenButton/>
    </HBox>
</HoverCanvas>
```

Pre-Built Hover-only Control Bar - Thumbnails

This special starter element is a horizontal control bar which displays a row of thumbnails; because it is inside a HoverCanvas, **it is only visible when the user moves their mouse**. It is designed to go at the bottom of the slideshow by default, but may be moved anywhere. The default buttons can be customized as needed. Buttons can also be added or removed.

The following image shows one of these control bars (with default configuration) in use within a slideshow:



Note: All the individual elements which comprise a starter element can be individually edited after the starter element has been added.

Equivalement Class File

This is a compound starter element made up of other elements; it has no individual associated class file.

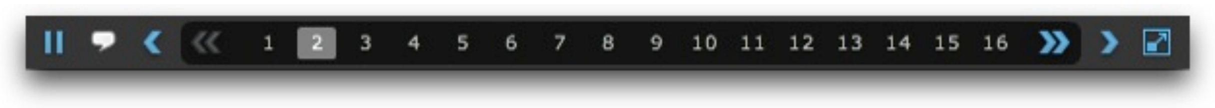
The Layout XML for the starter element is as follows:

```
<HoverCanvas left="0" right="0" bottom="0">
    <HBox left="0" right="0" bottom="0" paddingLeft="10" paddingRight="10" paddingTop="3"
paddingBottom="3" horizontalGap="10" backgroundColor="#000000" backgroundAlpha="0.8"
verticalAlign="middle">
        <PlayPauseButton/>
        <CaptionToggleButton/>
        <PreviousButton/>
        <Thumbnails width="100%" height="24"/>
        <NextButton/>
        <FullScreenButton/>
    </HBox>
</HoverCanvas>
```

Pre-Built Control Bar - NumberBar

This special starter element is a horizontal control bar which contains a NumberBar; it is always visible, regardless of user interaction (unlike its hover-only counterpart above). It is designed to go at the bottom of the slideshow by default, but may be moved anywhere. The default buttons can be customized as needed. Buttons can also be added or removed.

The following image shows one of these control bars (with default configuration) in use within a slideshow:



Note: All the individual elements which comprise a starter element can be individually edited after the starter element has been added.

Equivalement Class File

This is a compound starter element made up of other elements; it has no individual associated class file.

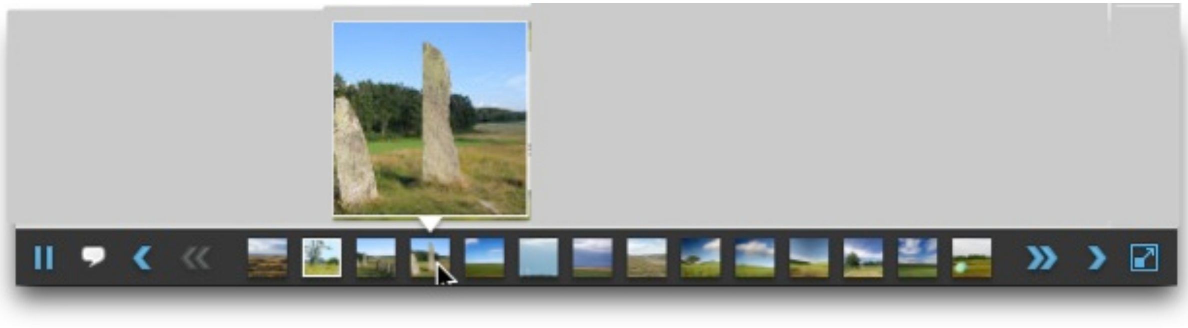
The Layout XML for the starter element is as follows:

```
<HBox left="0" right="0" bottom="0" paddingLeft="10" paddingRight="10" paddingTop="3" paddingBottom="3"
horizontalGap="10" backgroundColor="#000000" backgroundAlpha="0.8" verticalAlign="middle">
    <PlayPauseButton/>
    <CaptionToggleButton/>
    <PreviousButton/>
    <NumberBar width="100%"/>
    <NextButton/>
    <FullScreenButton/>
</HBox>
```

Pre-Built Control Bar - Thumbnails

This special starter element is a horizontal control bar which displays a row of thumbnails; it is always visible, regardless of user interaction (unlike its hover-only counterpart above). It is designed to go at the bottom of the slideshow by default, but may be moved anywhere. The default buttons can be customized as needed. Buttons can also be added or removed.

The following image shows one of these control bars (with default configuration) in use within a slideshow:



Note: All the individual elements which comprise a starter element can be individually edited after the starter element has been added.

Equivalement Class File

This is a compound starter element made up of other elements; it has no individual associated class file.

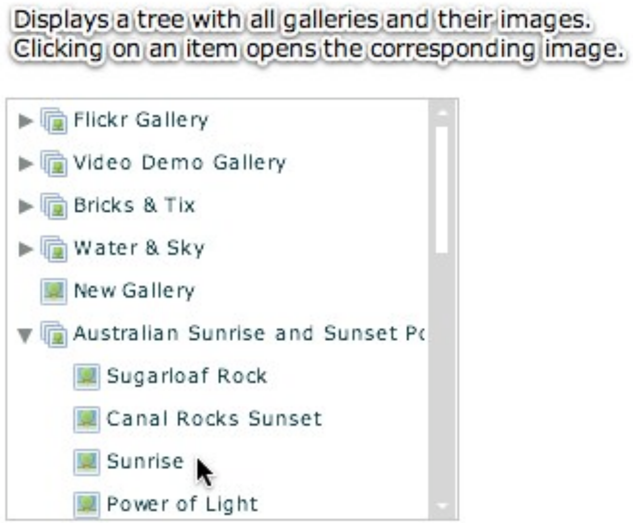
The Layout XML for the starter element is as follows:

```
<HBox left="0" right="0" bottom="0" paddingLeft="10" paddingRight="10" paddingTop="3" paddingBottom="3"
horizontalGap="10" backgroundColor="#000000" backgroundAlpha="0.8" verticalAlign="middle">
    <PlayPauseButton/>
    <CaptionToggleButton/>
    <PreviousButton/>
    <Thumbnails width="100%" height="24"/>
    <NextButton/>
    <FullScreenButton/>
</HBox>
```

ImageTree

The ImageTree element is useful when building portfolio-style presentations. It displays a hierachical tree of all galleries and their constituent images. When an item is selected in the tree, the corresponding slide is displayed.

The following image illustrates an ImageTree in use within a slideshow:



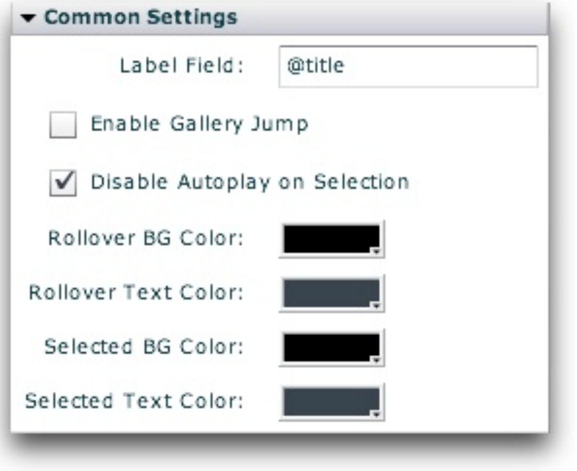
Displays a tree with all galleries and their images. Clicking on an item opens the corresponding image.

Important Usage Notes

It is often better to use the GalleryBar or GallerySelector elements to allow switching between galleries; the ImageTree is only for uniquely specialized instances.

Noteworthy Parameters

The Common Settings section of the Properties Panel contains several noteworthy parameters; they are described in the table below:



Common Settings

Label Field: @title

□ Enable Gallery Jump

☑ Disable Autoplay on Selection

Rollover BG Color: ■

Rollover Text Color: ■

Selected BG Color: ■

Selected Text Color: ■

Parameter name in Properties Panel	XML Property Name	Notes
Label Field	`labelField`	Specifies which gallery and image attribute (from the XML) should be displayed in the tree to describe each item. Usually the default value is best; it displays the gallery and image titles.
Enable Gallery Jump	`enableGalleryJump`	Specifies if the user can select a gallery folder in the tree to open that gallery. By default, the user must select an image item for anything to happen.

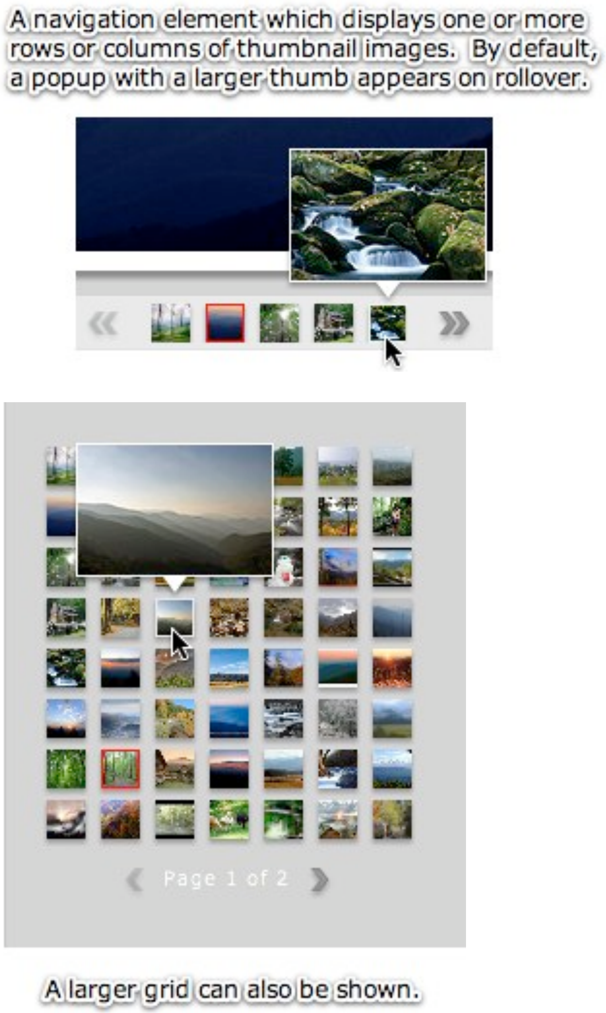
Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.ImageTree` class in the API Reference.

Thumbnails

The Thumbnails element is one of the most important elements. It displays a grid of all thumbnails in the current gallery. The element can be sized and oriented horizontally or vertically. By default, a tooltip with a larger thumbnail or more information is displayed when each thumbnail is hovered over.

The following images illustrate Thumbnails elements in use within a slideshow:



A navigation element which displays one or more rows or columns of thumbnail images. By default, a popup with a larger thumb appears on rollover.

A larger grid can also be shown.

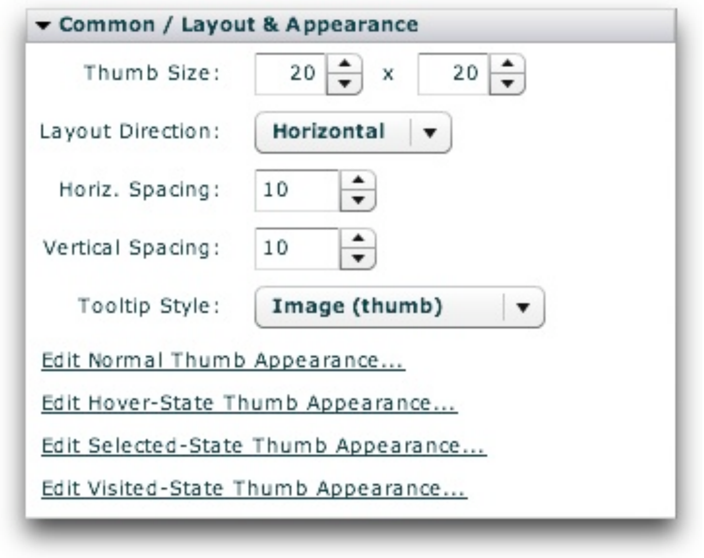
Important Usage Notes

Note the many paging control options shown below -- they provide a wide range of control over the paging / scrolling.

If you want a background, border, or drop shadow displayed behind the thumbnail grid, see the Border / Fill and Drop Shadow sections of the Properties Panel after selecting the element.

Noteworthy Parameters

The Common / Layout & Appearance section of the Properties Panel contains several noteworthy parameters; they are described in the table below:



Common / Layout & Appearance

Thumb Size: 20 x 20

Layout Direction: Horizontal

Horiz. Spacing: 10

Vertical Spacing: 10

Tooltip Style: Image (thumb)

Edit Normal Thumb Appearance...

Edit Hover-State Thumb Appearance...

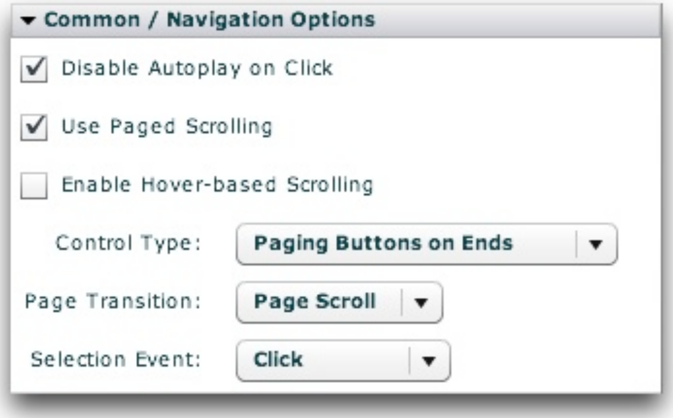
Edit Selected-State Thumb Appearance...

Edit Visited-State Thumb Appearance...

Parameter name in Properties Panel	XML Property Name	Notes
Thumb Size	`tileWidth`	Specifies (in pixels) the width of each thumbnail
	`tileHeight`	Specifies (in pixels) the height of each thumbnail
Horizontal Spacing	`horizontalGap`	Specifies (in pixels) the horizontal spacing between each thumbnail
Vertical Spacing	`verticalGap`	Specifies (in pixels) the vertical spacing between each thumbnail
Tooltip Style	`tooltipStyle`	Specifies the type of popup tooltip shown when each thumb is hovered over. Possible values include: ▪ **Image** - Displays a larger version of the thumbnail image. ▪ **Text** - Displays the image's title and [optionally] description. If this setting is used, the Text Tooltip Options section of the Properties Panel provides additional settings (see below for more information). ▪ **None** - Disables the tooltip.
Edit Normal	*Controls the*	Opens a popup dialog where you can manipulate the appearance of thumbnails in their standard 'up' state,

Thumb Appearance...	***<Thumb>** style selector settings*	including border / matte, opacity, color manipulations, and much more.
Edit Hover-State Thumb Appearance...	*Controls the **<ThumbHover>** style selector settings*	Opens a popup dialog where you can manipulate the appearance of thumbnails when they are hovered-over, including border / matte, opacity, color manipulations, and much more.
Edit Selected-State Thumb Appearance...	*Controls the **<ThumbSelected>** style selector settings*	Opens a popup dialog where you can manipulate the appearance of thumbnails in their selected state (when they represent the currently selected slide), including border / matte, opacity, color manipulations, and much more. These settings override the Normal and Visited state, but are overridden by the Hover state.
Edit Visited-State Thumb Appearance...	*Controls the **<ThumbVisited>** style selector settings*	Opens a popup dialog where you can manipulate the appearance of thumbnails in their visited state (when they represent slides which have been already viewed), including border / matte, opacity, color manipulations, and much more. These settings override the Normal state, but are overridden by the Hover and Visited states.

The Common / Navigation Options section of the Properties Panel also contains many important settings; they are described in the table below:



Common / Navigation Options

☑ Disable Autoplay on Click

☑ Use Paged Scrolling

☐ Enable Hover-based Scrolling

Control Type: Paging Buttons on Ends

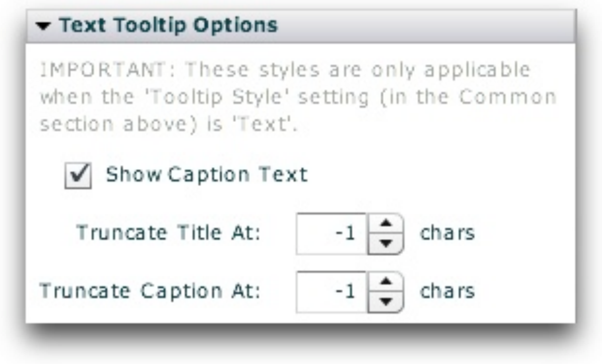
Page Transition: Page Scroll

Selection Event: Click

Parameter name in Properties Panel	XML Property Name	Notes
Use Paged Scrolling	`usePagedScrolling`	Specifies if scrolling should be page-based or continuous.
Enable Hover-	`hoverScrollEnabled`	Specifies if hover-based scrolling is enabled (only works if 'Use Paged Scrolling' is disabled). If enabled

based Scrolling		and 'Use Paged Scrolling' is disabled, the user can scroll simply by mousing over the element and moving the mouse towards one end or the other.
Control Type	`pagingStyle`	In the default state (when no custom PagingControls descriptor has been specified), specifies the type and location of paging controls shown.
Page Transition	`transitionType`	Specifies the type of transition effect used between each page when page-based scrolling is enabled.
Selection Event	`selectionEvent`	Specifies whether a tile will become selected when the user clicks it, or when they simply mouse over it. Usually best to leave the default value of Click.

The Text Tooltip Options section of the Properties Panel contains settings which control the appearance of the tooltip if the Text tooltip option was chosen for the 'Tooltip Style' setting. If the Image or None setting is used, the settings in this panel are not applicable. The table below describes the options:



▾ Text Tooltip Options

IMPORTANT: These styles are only applicable when the 'Tooltip Style' setting (in the Common section above) is 'Text'.

☑ Show Caption Text

Truncate Title At: -1 chars

Truncate Caption At: -1 chars

Parameter name in Properties Panel	XML Property Name	Notes
Show Caption Text	`tooltipShowCaption`	Specifies if the caption text should be shown along with the title text in the tooltip. By default, both are shown; if this setting is disabled, only the title text will be shown.
Truncate Title At	`tooltipTruncateTitleAt`	Specifies the maximum length of the title text displayed in the tooltip. If the length exceeds the specified value, the title text will be truncated with an ellipsis (...). The default value, `-1`, means that truncation should never occur, regardless of length.
Truncate Caption At	`tooltipTruncateCaptionAt`	Specifies the maximum length of the caption text displayed in the tooltip. If the length exceeds the specified value, the caption text will be truncated with an ellipsis (...). The default value, `-1`, means that truncation should never occur, regardless of length.

Equivalement Class File

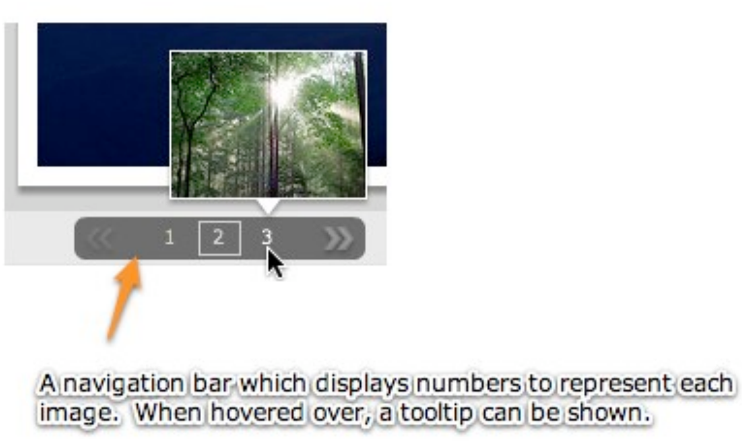
For detailed reference on this element, see the `com.dwuser.ss4.ui.Thumbnails` class in the API Reference.

For the thumbnail classes, see `com.dwuser.ss4.ui.Thumb` and `com.dwuser.ss4.pseudoStyleClasses.ThumbHover` and `com.dwuser.ss4.pseudoStyleClasses.ThumbSelected` and `com.dwuser.ss4.pseudoStyleClasses.ThumbVisited` .

NumberBar

The NumberBar element is analogous to the JumpBar in version 3. It displays a grid (or single row/column) of numbers with one to represent each slide in the current gallery. The element can be sized and oriented horizontally or vertically. By default, a tooltip with a thumbnail or more information is displayed when each number is hovered over.

The following image illustrates a NumberBar element in use within a slideshow:



A navigation bar which displays numbers to represent each image. When hovered over, a tooltip can be shown.

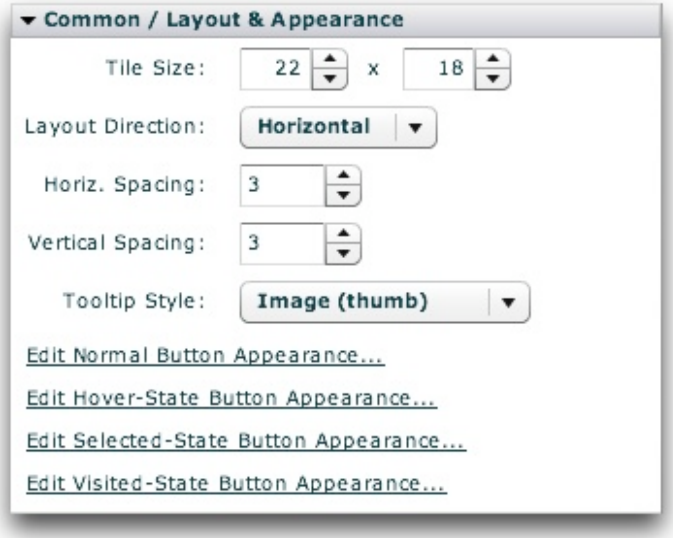
Important Usage Notes

Note the many paging control options shown below -- they provide a wide range of control over the paging / scrolling.

If you want a background, border, or drop shadow displayed behind the NumberBar, see the Border / Fill and Drop Shadow sections of the Properties Panel after selecting the element.

Noteworthy Parameters

The Common / Layout & Appearance section of the Properties Panel contains several noteworthy parameters; they are described in the table below:



Common / Layout & Appearance

Tile Size: 22 x 18

Layout Direction: Horizontal

Horiz. Spacing: 3

Vertical Spacing: 3

Tooltip Style: Image (thumb)

Edit Normal Button Appearance...

Edit Hover-State Button Appearance...

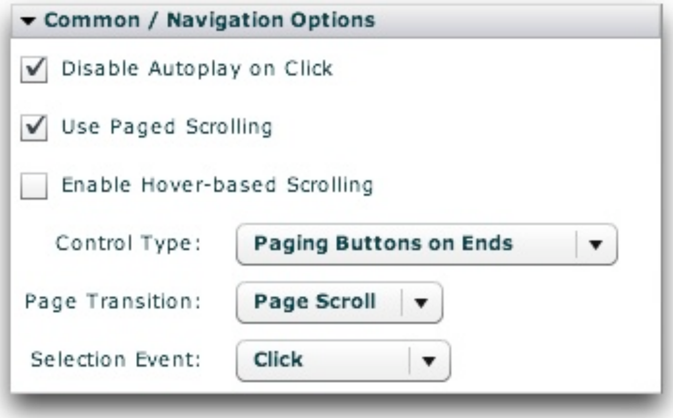
Edit Selected-State Button Appearance...

Edit Visited-State Button Appearance...

Parameter name in Properties Panel	XML Property Name	Notes
Tile Size	`tileWidth`	Specifies (in pixels) the width of each tile.
	`tileHeight`	Specifies (in pixels) the height of each tile.
Horizontal Spacing	`horizontalGap`	Specifies (in pixels) the horizontal spacing between each tile.
Vertical Spacing	`verticalGap`	Specifies (in pixels) the vertical spacing between each tile.
Tooltip Style	`tooltipStyle`	Specifies the type of popup tooltip shown when each number is hovered over. Possible values include: ▪ **Image** - Displays a thumbnail image. ▪ **Text** - Displays the image's title and [sometimes truncated] description. If this setting is used, the Text Tooltip Options section of the Properties Panel provides additional settings (see below for more information). ▪ **None** - Disables the tooltip.
Edit Normal Button	*Controls the* ***<NumberBarTile>*** *style*	Opens a popup dialog where you can manipulate the appearance of the tiles in their standard 'up' state, including border / matte, opacity, color manipulations, and much more.

Appearance...	*selector settings*	
Edit Hover-State Button Appearance...	*Controls the* ***<NumberBarTileHover>*** *style selector settings*	Opens a popup dialog where you can manipulate the appearance of the tiles when they are hovered-over, including border / matte, opacity, color manipulations, and much more.
Edit Selected-State Button Appearance...	*Controls the* ***<NumberBarTileSelected>*** *style selector settings*	Opens a popup dialog where you can manipulate the appearance of the tiles in their selected state (when they represent the currently selected slide), including border / matte, opacity, color manipulations, and much more. These settings override the Normal and Visited state, but are overridden by the Hover state.
Edit Visited-State Button Appearance...	*Controls the* ***<NumberBarTileVisited>*** *style selector settings*	Opens a popup dialog where you can manipulate the appearance of the tiles in their visited state (when they represent slides which have been already viewed), including border / matte, opacity, color manipulations, and much more. These settings override the Normal state, but are overridden by the Hover and Visited states.

The Common / Navigation Options section of the Properties Panel also contains many important settings; they are described in the table below:



Common / Navigation Options

☑ Disable Autoplay on Click

☑ Use Paged Scrolling

☐ Enable Hover-based Scrolling

Control Type: Paging Buttons on Ends

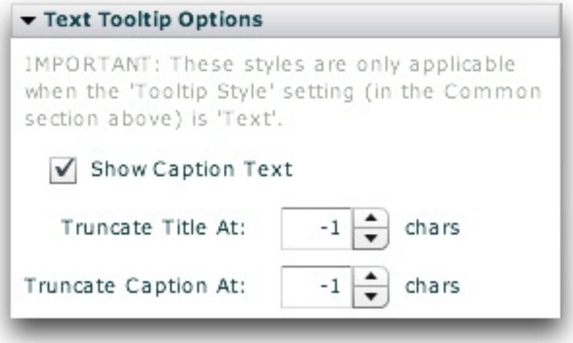
Page Transition: Page Scroll

Selection Event: Click

Parameter name in Properties Panel	XML Property Name	Notes
Use Paged Scrolling	`usePagedScrolling`	Specifies if scrolling should be page-based or continuous.
Enable Hover-based Scrolling	`hoverScrollEnabled`	Specifies if hover-based scrolling is enabled (only works if 'Use Paged Scrolling' is disabled). If enabled and 'Use Paged Scrolling' is disabled, the user can scroll simply by mousing over the element and moving

		the mouse towards one end or the other.
Control Type	`pagingStyle`	In the default state (when no custom PagingControls descriptor has been specified), specifies the type and location of paging controls shown.
Page Transition	`transitionType`	Specifies the type of transition effect used between each page when page-based scrolling is enabled.
Selection Event	`selectionEvent`	Specifies whether a tile will become selected when the user clicks it, or when they simply mouse over it. Usually best to leave the default value of Click.

The Text Tooltip Options section of the Properties Panel contains settings which control the appearance of the tooltip if the Text tooltip option was chosen for the 'Tooltip Style' setting. If the Image or None setting is used, the settings in this panel are not applicable. The table below describes the options:



▾ Text Tooltip Options

IMPORTANT: These styles are only applicable when the 'Tooltip Style' setting (in the Common section above) is 'Text'.

☑ Show Caption Text

Truncate Title At: -1 chars

Truncate Caption At: -1 chars

Parameter name in Properties Panel	XML Property Name	Notes
Show Caption Text	`tooltipShowCaption`	Specifies if the caption text should be shown along with the title text in the tooltip. By default, both are shown; if this setting is disabled, only the title text will be shown.
Truncate Title At	`tooltipTruncateTitleAt`	Specifies the maximum length of the title text displayed in the tooltip. If the length exceeds the specified value, the title text will be truncated with an ellipsis (...). The default value, `-1`, means that truncation should never occur, regardless of length.
Truncate Caption At	`tooltipTruncateCaptionAt`	Specifies the maximum length of the caption text displayed in the tooltip. If the length exceeds the specified value, the caption text will be truncated with an ellipsis (...). The default value, `-1`, means that truncation should never occur, regardless of length.

Equivalement Class File

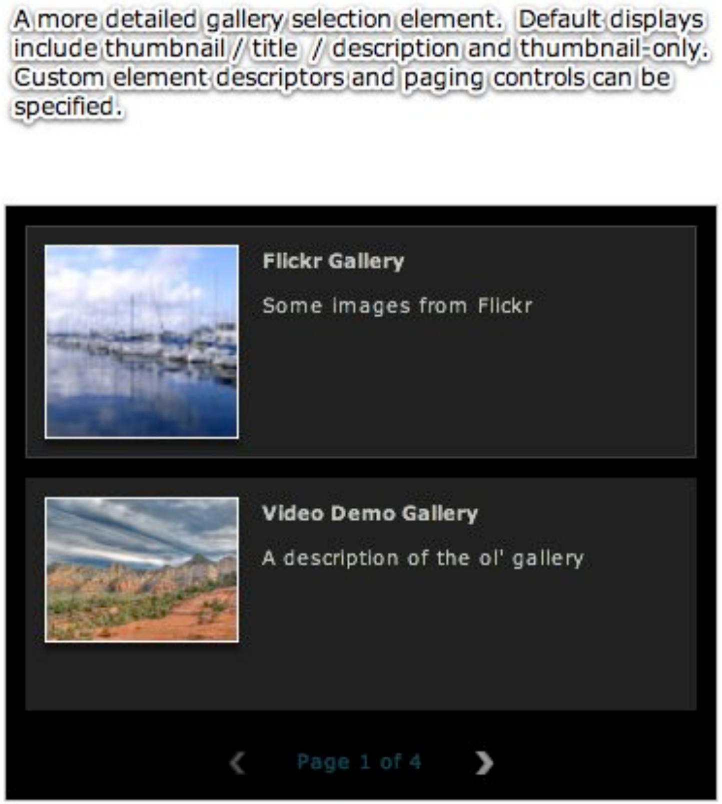
For detailed reference on this element, see the `com.dwuser.ss4.ui.NumberBar` class in the API Reference.

For the tile (number button) classes, see `com.dwuser.ss4.ui.NumberBarTile` and `com.dwuser.ss4.pseudoStyleClasses.NumberBarTileHover` and `com.dwuser.ss4.pseudoStyleClasses.NumberBarTileSelected` and `com.dwuser.ss4.pseudoStyleClasses.NumberBarTileVisited` .

GallerySelector

The GallerySelector element provides a grid of all image galleries in the slideshow (with automatic pagination if needed). Two default display types are supported -- thumbnail+title+description or thumbnail-only. When one of the items is clicked on, the corresponding gallery is opened. It is analogous to the Gallery element in version 3.

The following image illustrates a GallerySelector in use within a slideshow:



A more detailed gallery selection element. Default displays include thumbnail / title / description and thumbnail-only. Custom element descriptors and paging controls can be specified.

Important Usage Notes

For a more compact display, consider using the GalleryBar element for gallery selection.

The Display Type parameter described below is very useful; it can be used to create an images-only gallery selection screen.

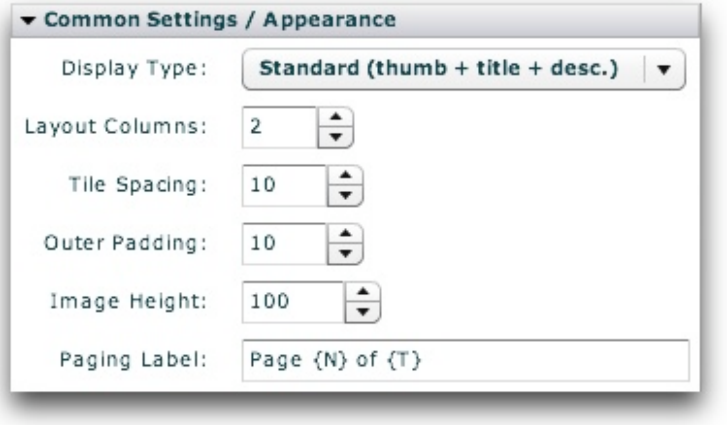
By default, when an item in the GallerySelector is clicked the corresponding gallery is opened. If the `hideOnSelection` style parameter is left at its default value of `true`, the actual GallerySelector element is hidden as well. (You can set this parameter to `false` to disable this behavior.) If you want users to be able to select a gallery again later, you'll need to:

- Specify a unique ID reference value for the GallerySelector
- Add a static Button element which has the following click event specified: *`gallerySelectorID`*`.visible = true`

If you've enabled showing/hiding the GallerySelector, you may also want to apply a show and hide effect (such as a fade) in the Effects section of the Properties Panel.

Noteworthy Parameters

The Common Settings section of the Properties Panel contains several noteworthy parameters; they are described in the table below:



Common Settings / Appearance

Display Type: Standard (thumb + title + desc.)

Layout Columns: 2

Tile Spacing: 10

Outer Padding: 10

Image Height: 100

Paging Label: Page {N} of {T}

Parameter name in Properties Panel	XML Property Name	Notes
Display Type	`displayType`	Specifies the type of tile which is created to represent each gallery. Supported default values are: - **Standard** - As shown in the example image above, displays the gallery's thumbnail, title and description.

		- **Images / Thumb Only** - Only displays the gallery thumbnail. The Image Height parameter below affects the tile size.
Layout Columns	`columns`	Specifies the number of columns in the layout grid.
Tile Spacing	`spacing`	Specifies, in pixels, the spacing between each gallery tile.
Outer Padding	`outerSpacing`	Specifies, in pixels, the spacing between the designated bounds of the GallerySelector and the actual layout grid.
Image Height	`imageHeight`	Only applicable when using the 'Images / Thumb Only' display type above. In this case, specifies the allotted vertical image space (in pixels).
Paging Label	`statusLabel`	If there is more than one page of galleries, specifies the paging label displayed. Use `{N}` to represent the current page, and `{T}` to represent the total number of pages.

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.GallerySelector` class in the API Reference.

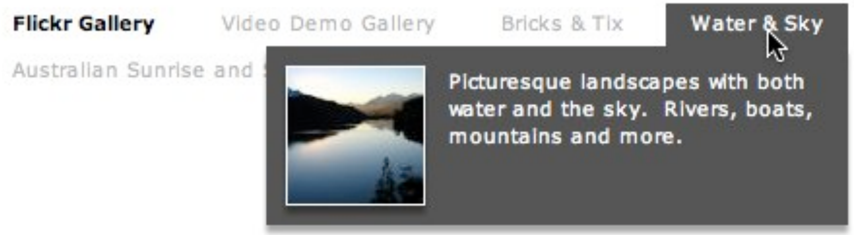
GalleryBar

The GalleryBar provides a compact bar which allows gallery selection. It is designed to be on the same screen (layout view) as the main image display, removing any need for separate Images and Gallery Selection layout views. It rests unobtrusively at the top of the screen, providing more information only when hovered over.

The title of each gallery is displayed; when hovered over, more information is displayed in a tooltip. If needed, the items wrap to additional rows once the first row is filled.

The following image illustrates a GallerySelector in use within a slideshow:

A compact selection bar which has a button for each gallery. When a button is hovered over, a tooltip with gallery information is by default shown.



Picturesque landscapes with both water and the sky. Rivers, boats, mountains and more.

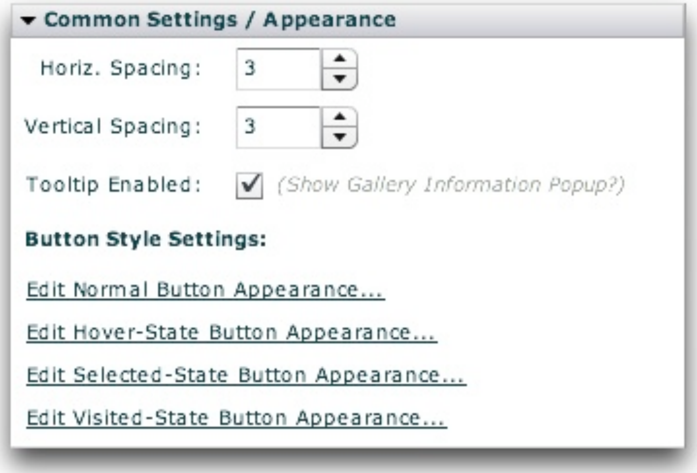
Important Usage Notes

For a more expanded display, you can use the GallerySelector element for gallery selection.

When inserted, the element by default has a width `100%`, which may make it seem to behave differently than other elements in Design View. You can change this if needed.

Noteworthy Parameters

The Common Settings section of the Properties Panel contains several noteworthy parameters; they are described in the table below:



▾ Common Settings / Appearance

Horiz. Spacing: 3

Vertical Spacing: 3

Tooltip Enabled: ☑ *(Show Gallery Information Popup?)*

Button Style Settings:

Edit Normal Button Appearance...

Edit Hover-State Button Appearance...

Edit Selected-State Button Appearance...

Edit Visited-State Button Appearance...

Parameter name in Properties Panel	XML Property Name	Notes
Horizontal Spacing	`horizontalGap`	Specifies, in pixels, the horizontal gap (spacing) between each button on the bar.
Vertical Spacing	`verticalGap`	Specifies, in pixels, the vertical gap (spacing) between each button on the bar.
Tooltip Enabled	`tooltipEnabled`	Specifies if the popup tooltip with gallery information is shown when the user hovers over a button on the bar.
Edit Normal Button Appearance...	*Controls the **<GalleryBarButton>** style selector settings*	Opens a popup dialog where you can manipulate the appearance of the buttons in their standard 'up' state, including border / matte, opacity, color manipulations, and much more.
Edit Hover-State Button Appearance...	*Controls the **<GalleryBarHover>** style selector settings*	Opens a popup dialog where you can manipulate the appearance of the buttons when they are hovered-over, including border / matte, opacity, color manipulations, and much more.
Edit Selected-State Button Appearance...	*Controls the **<GalleryBarSelected>** style selector settings*	Opens a popup dialog where you can manipulate the appearance of the buttons in their selected state (when they represent the currently selected gallery), including border / matte, opacity, color manipulations, and much more. These settings override the Normal and Visited state, but are overridden by the Hover state.
Edit Visited-State Button Appearance...	*Controls the **<GalleryBarVisited>** style selector settings*	Opens a popup dialog where you can manipulate the appearance of the buttons in their visited state (when they represent galleries which have been already viewed), including border / matte, opacity, color manipulations, and much more. These settings override the Normal state, but are overridden by the Hover and Visited states.

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.GalleryBar` class in the API Reference.

DynamicScrollableText

The DynamicScrollableText element displays scrollable text which dynamically changes based on the currently selected gallery and image (such as the image title or caption). If the text doesn't fit in the available/allowed area, a scrollbar automatically appears. When the text changes, it automatically fades between the old and new values. The DynamicScrollableText element is identical to the DynamicText element except that, if the text doesn't fit in the available area, the extra text can be scrolled into view on this element whereas it is hidden on the other element.

By default, the scrolling on the DynamicScrollableText element can be triggered with the keyboard shortcut of '<' (up) or '>' (down).

The most important property on this element is the Displayed Text value (the `text` attribute in the XML), which controls the text displayed. Note - you can use the `htmlText` property instead of `text` if you wish to display HTML text. The Text Formatting section of the Properties Panel controls the formatting of the element's text.

About Replacement Values

The DynamicScrollableText element has support for replacement values in its label. Replacement values are special string values which are automatically replaced based on the current gallery and image's information. The following gallery-related replacement values are available:

- `{gallery.index}` or `{gallery.id}` - This string is replaced with the 1-base index of the current gallery. For example, "This is gallery #{gallery.id}" would become "This is gallery #1" while the first gallery was open, then subsequently "This is gallery #2", etc.
- `{gallery.total}` - This string is replaced with the total number of galleries. For example, "This is gallery {gallery.id}/{gallery.total}" would become "This is gallery 1/10" if there were ten galleries and the first gallery was open.
- `{gallery.index0}` or `{gallery.id0}` - This string is replaced with the **0**-base index of the current image. This property is generally only used by advanced users; for most purposes, the 1-base `{gallery.index}` value is appropriate.
- `{gallery.uniqID}` - This string is replaced with the uniqID index of the current gallery. This can be applicable when working with permalinking, and will not be affected by randomization. This property is generally only used by advanced users.
- `{gallery.rawID}` - This string is replaced with the `id` attribute defined on the actual gallery node; this is the value set in the wizard in the Gallery Linkage ID field. This is generally a property only used by advanced developers.
- `{gallery.*someAtt*}` - This string is replaced with the corresponding XML attribute of the current gallery's XML node. For example, "Current gallery: {gallery.title}" would become "Current gallery: Flowers and Sunshine" if the gallery's title was defined like this: `<gallery title="Flowers and Sunshine" ... >`. You can use the standard attributes (such as title and description), or any custom attribute which has been defined.

The following image-related replacement values are available:

- `{image.index}` or `{image.id}` - This string is replaced with the 1-base index of the current image. For example, "This is image #{image.id}" would become "This is image #1" while the first image was open, then subsequently "This is image #2", etc.
- `{image.total}` - This string is replaced with the total number of images in the current gallery. For example, "This is image {image.id}/{image.total}" would become "This is image 1/10" if there were ten images in the gallery and the first image was open.
- `{image.link}` - This string is replaced with the current image's link value (if one has been defined). For example, "Purchase from: {image.link}" would become "Purchase from: http://www.mysite.com/" if the image's link was defined like this: `<img link="http://www.mysite.com/" ... />`.
- `{image.title}` - This string is replaced with the current image's title value (if one has been defined). For example, "Current image: {image.title}" would become "Current image: Daffodil" if the image's title was defined like this: `<img ... title="Daffodil" ... />`.
- `{image.caption}` - This string is replaced with the current image's caption value (if one has been defined). For example, "About: {image.caption}" would become "About: Taken two weeks ago on holiday in South Africa" if the image's caption was defined like this: `<img ... caption="Taken`

two weeks ago on holiday in South Africa" ... />.

- {image.*someOtherAtt*} - This string is replaced with the corresponding XML attribute of the current image's XML node. For example, "Current image: {image.title}" would become "Current image: Blue Orchid" if the image's title was defined like this: <img title="Blue Orchid" ... >. You can use the standard attributes (such as title and caption), or any custom attribute which has been defined. In other words, {image.someValue} would be replaced with <img ... someValue="Special attribute value!" ... />
- **[Metadata support]** {metadata.*property*} - This string is replaced with various available metadata parameters that are read from the loaded images. Click here to learn more about metadata integration. The available *property* values include {metadata.creator} (corresponds to the Creator metadata property), {metadata.title} (corresponds to the Title metadata property), and {metadata.description} (corresponds to the Description metadata property). If a requested metadata property doesn't exist on an image, or if the metadata isn't available because the image hasn't fully loaded yet, a default value can be specified using the following format: {metadata.property**|Default Value Here**} . After the property value, a pipe character denotes the start of the default string. For example, the following value would display *"Loading..."* until the metadata title became available: {metadata.title|Loading...} .

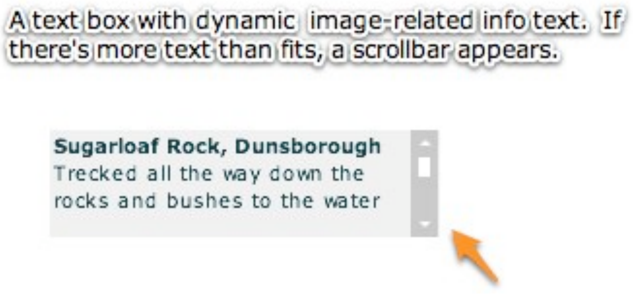
Other less-common image-related replacement values are available too:

- {image.index0} or {image.id0} - This string is replaced with the **0**-base index of the current image. This is generally a property used only by advanced users; for most purposes, the 1-base {image.index} value is appropriate.
- {image.uniqID} - This string is replaced with the uniqID index of the current image. This can be used for permalinking, and will not be affected by randomization. This is generally a property only used by advanced users.
- {image.rawID} - This string is replaced with the id attribute defined on the actual image node; this is the value set in the wizard in the Image Linkage ID field. This is generally a property only used by advanced users.
- {image.src} - This string is replaced with the current image's src value. This usually represents just the file name of the image; if you want to get the entire image path, use the {image.srcFull} replacement value.
- {image.srcFull} - This string is replaced with the current image's full image path (use {image.src} for just the shorter path / file name).
- {image.thumbFull} - This string is replaced with the current image's full thumb path (use {image.thumb} for just the shorter path / file name)

For all replacement values, you can use the following format to specify a default value: {image.property**|Default Value Here**} . Follow the property name with a pipe character (|) and the default value before the closing curly brace.

Note that you can use multiple replacement values in the same string.

Here is an example of a DynamicScrollableText element in use within a slideshow:



A text box with dynamic image-related info text. If there's more text than fits, a scrollbar appears.

Sugarloaf Rock, Dunsborough
Trecked all the way down the
rocks and bushes to the water

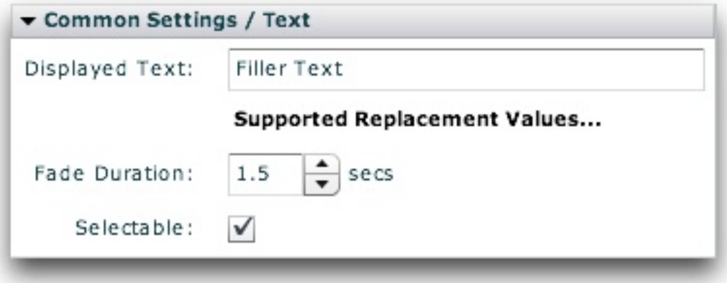
Important Usage Notes

This element is related to the DynamicLinkButton, Caption, and DynamicText elements. If you want a "pre-baked" display of the image caption and title, use the Caption element. On the other hand, this element can be used to display only one (caption or title) and customize the formatting on each separately.

If you want to display just the caption or title, you can also use the Plain Caption Text or Plain Title Text starter elements in the Special Starter Elements section.

Noteworthy Parameters

The Common Settings section of the Properties Panel contains several noteworthy parameters; they are described in the table below:



▾ Common Settings / Text

Displayed Text: Filler Text

Supported Replacement Values...

Fade Duration: 1.5 secs

Selectable: ☑

Parameter name in Properties Panel	XML Property Name	Notes
Displayed Text	`text`	The text value displayed. Replacement values are supported. For more information about the replacement values, click the 'Supported Replacement Values...' link.
Fade Duration	`fadeTime`	Specifies the duration of the fade effect which is used to transition between different text values.
Selectable	`selectable`	Specifies if the text can be selected.

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.DynamicScrollableText` class in the API Reference.

DynamicLinkButton

The DynamicLinkButton element creates a clickable link button whose text and link dynamically change based on the current selected gallery and image.

The two important properties on this element are the Label Text and Link URL properties. They control (respectively) the label displayed, and the link which is opened when the button is clicked.

The special property of the DynamicLinkButton element is its support of replacement values in the label and link. These are values which dynamically change based on the current image.

About Replacement Values

The DynamicLinkButton element has support for replacement values in its label. Replacement values are special string values which are automatically replaced based on the current gallery and image's information. The following gallery-related replacement values are available:

<

- `{gallery.index}` or `{gallery.id}` - This string is replaced with the 1-base index of the current gallery. For example, "This is gallery #{gallery.id}" would become "This is gallery #1" while the first gallery was open, then subsequently "This is gallery #2", etc.
- `{gallery.total}` - This string is replaced with the total number of galleries. For example, "This is gallery {gallery.id}/{gallery.total}" would become "This is gallery 1/10" if there were ten galleries and the first gallery was open.
- `{gallery.index0}` or `{gallery.id0}` - This string is replaced with the **0**-base index of the current image. This property is generally only used by advanced users; for most purposes, the 1-base `{gallery.index}` value is appropriate.
- `{gallery.uniqID}` - This string is replaced with the uniqID index of the current gallery. This can be applicable when working with permalinking, and will not be affected by randomization. This property is generally only used by advanced users.
- `{gallery.rawID}` - This string is replaced with the `id` attribute defined on the actual gallery node; this is the value set in the wizard in the Gallery Linkage ID field. This is generally a property only used by advanced developers.
- `{gallery.someAtt}` - This string is replaced with the corresponding XML attribute of the current gallery's XML node. For example, "Current gallery: {gallery.title}" would become "Current gallery: Flowers and Sunshine" if the gallery's title was defined like this: `<gallery title="Flowers and Sunshine" ... >`. You can use the standard attributes (such as title and description), or any custom attribute which has been defined.

The following image-related replacement values are available:

- `{image.index}` or `{image.id}` - This string is replaced with the 1-base index of the current image. For example, "This is image #{image.id}" would become "This is image #1" while the first image was open, then subsequently "This is image #2", etc.
- `{image.total}` - This string is replaced with the total number of images in the current gallery. For example, "This is image {image.id}/{image.total}" would become "This is image 1/10" if there were ten images in the gallery and the first image was open.
- `{image.link}` - This string is replaced with the current image's link value (if one has been defined). For example, "Purchase from: {image.link}" would become "Purchase from: http://www.mysite.com/" if the image's link was defined like this: `<img link="http://www.mysite.com/" ... />`.
- `{image.title}` - This string is replaced with the current image's title value (if one has been defined). For example, "Current image: {image.title}" would become "Current image: Daffodil" if the image's title was defined like this: `<img ... title="Daffodil" ... />`.
- `{image.caption}` - This string is replaced with the current image's caption value (if one has been defined). For example, "About: {image.caption}" would become "About: Taken two weeks ago on holiday in South Africa" if the image's caption was defined like this: `<img ... caption="Taken two weeks ago on holiday in South Africa" ... />`.
- `{image.`*`someOtherAtt`*`}` - This string is replaced with the corresponding XML attribute of the current image's XML node. For example, "Current image: {image.title}" would become "Current image: Blue Orchid" if the image's title was defined like this: <img title="Blue Orchid" ... >. You can use the standard attributes (such as title and caption), or any custom attribute which has been defined. In other words, {image.someValue} would be replaced with `<img ... someValue="Special attribute value!" ... />`
- **[Metadata support]** `{metadata.`*`property`*`}` - This string is replaced with various available metadata parameters that are read from the loaded images. Click here to learn more about metadata integration. The available *property* values include `{metadata.creator}` (corresponds to the Creator metadata property), `{metadata.title}` (corresponds to the Title metadata property), and `{metadata.description}` (corresponds to the Description metadata property). If a requested metadata property doesn't exist on an image, or if the metadata isn't available because the image hasn't fully loaded yet, a default value can be specified using the following format: `{metadata.property`**`|Default Value Here`**`}` . After the property value, a pipe character denotes the start of the default string. For example, the following value would display *"Loading..."* until the metadata title became available: `{metadata.title|Loading...}` .

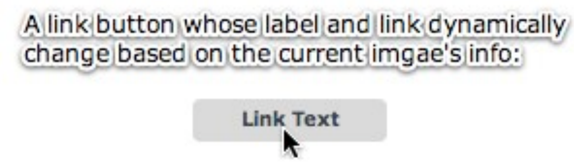
Other less-common image-related replacement values are available too:

- `{image.index0}` or `{image.id0}` - This string is replaced with the **0**-base index of the current image. This is generally a property used only by advanced users; for most purposes, the 1-base `{image.index}` value is appropriate.
- `{image.uniqID}` - This string is replaced with the uniqID index of the current image. This can be used for permalinking, and will not be affected by randomization. This is generally a property only used by advanced users.
- `{image.rawID}` - This string is replaced with the id attribute defined on the actual image node; this is the value set in the wizard in the Image Linkage ID field. This is generally a property only used by advanced users.
- `{image.src}` - This string is replaced with the current image's src value. This usually represents just the file name of the image; if you want to get the entire image path, use the {image.srcFull} replacement value.
- `{image.srcFull}` - This string is replaced with the current image's full image path (use {image.src} for just the shorter path / file name).
- `{image.thumbFull}` - This string is replaced with the current image's full thumb path (use {image.thumb} for just the shorter path / file name)

For all replacement values, you can use the following format to specify a default value: `{image.property`**`|Default Value Here`**`}`. Follow the property name with a pipe character (`|`) and the default value before the closing curly brace.

Note that you can use multiple replacement values in the same string.

Here is an example of a DynamicLinkButton element in use within a slideshow:



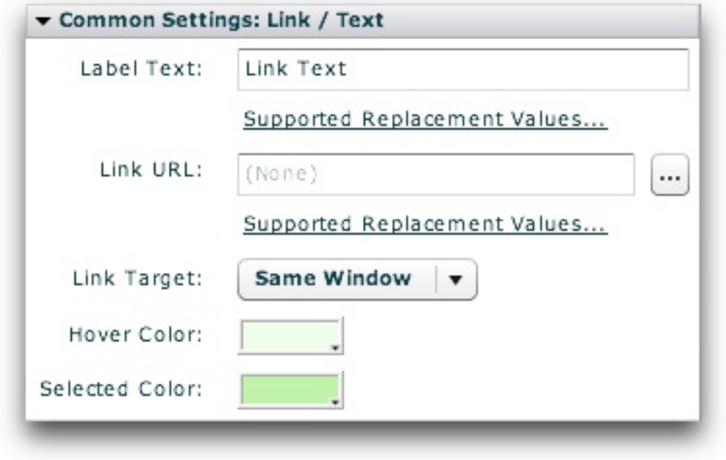
A link button whose label and link dynamically change based on the current imgae's info:

Important Usage Notes

This element is related to the DynamicScrollableText, Caption, and DynamicText elements.

Noteworthy Parameters

The Common Settings section of the Properties Panel contains several noteworthy parameters; they are described in the table below:



▾ Common Settings: Link / Text

Label Text: Link Text

Supported Replacement Values...

Link URL: (None)

...
Supported Replacement Values...

Link Target: Same Window ▾

Hover Color:

Selected Color:

Parameter name in Properties Panel	XML Property Name	Notes
Label Text	`label`	The label displayed on the link button. Replacement values are supported. For more information about the replacement values, click the 'Supported Replacement Values...' link.

Link URL	`link`	The link which is opened when the button is clicked. Replacement values are supported. For more information about the replacement values, click the 'Supported Replacement Values...' link.
Link Target	`linkTarget`	Specifies the window in which the link will be opened when the button is clicked.
Hover Color	`rollOverColor`	Specifies the color of the button's background/highlight when the user hovers over it.
Selected Color	`selectionColor`	Specifies the color of the button's background/highlight when the user clicks it.

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.DynamicLinkButton` class in the API Reference.

Caption

The Caption element displays both the current image's title and it's current caption value. It is designed to be a helpful starter element which can be customized further if you want different text displayed, or different formatting used to display the text.

If you wish only to change the caption formatting, you can use the Text Formatting section of the Properties Panel. However, this will not affect the title formatting. To customize the title value, you must use a custom descriptor as shown below.

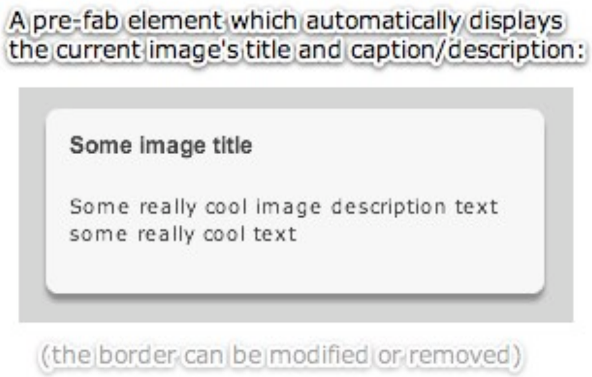
The Caption element has the special features of default show/hide and resize effects, as well as automatic hiding when no image caption has been defined.

By utilizing a special feature known as Default Child Descriptors, the Caption element by default has the following child nodes inside of it (if none are explicitly specified):

```
<Caption>
    <DynamicScrollableText htmlText="{image.title}" width="100%" fontSize="12" fontFamily="Arial"
fontWeight="bold" />
    <DynamicScrollableText maxHeight="200" htmlText="{image.caption}" width="100%" />
</Caption>
```

However, if you explicitly specify any child nodes, this default descriptor will not be used. Thus, when you want to further customize the Caption element, you can copy and paste the above code within the `<Caption ... > ... </Caption>` tags in Source View and make modifications as needed. See the DynamicScrollableText and DynamicText elements to learn more about how to modify those XML nodes.

Here is an example of a Caption element in use within a slideshow:



A pre-fab element which automatically displays the current image's title and caption/description:

Some image title

Some really cool image description text some really cool text

(the border can be modified or removed)

Important Usage Notes

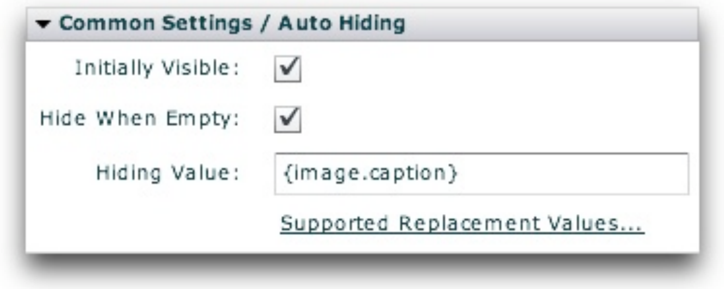
This element is related to the DynamicLinkButton, DynamicScrollableText, and DynamicText elements. If you want a "pre-baked" display of the image caption and title, use the Caption element. On the other hand, the DynamicScrollableText and DynamicText elements can be used to display only one (caption or title) and customize the formatting on each separately.

You can use the CaptionToggleButton element (see Slideshow Control Buttons) to give the user the option of toggling the caption on and off.

If you want to display just the caption or title, you can also use the Plain Caption Text or Plain Title Text starter elements in the Special Starter Elements section.

Noteworthy Parameters

The Common Settings section of the Properties Panel contains several noteworthy parameters; they are described in the table below:



Common Settings / Auto Hiding

Initially Visible: ☑

Hide When Empty: ☑

Hiding Value: {image.caption}

Parameter name in Properties	XML Property Name	Notes

Panel		
Initially Visible	`outerVisible`	Specifies if the caption element should initially be visible or hidden. A CaptionToggleButton can be used to allow hiding or revealing of the Caption element.
Hide When Empty	`hideWhenEmpty`	Specifies whether the caption should automatically hide itself when the 'Hiding Value' specified evaluates to an empty string.
Hiding Value	`hideWhenEmptyValue`	Specifies the string, which when evaluated to be empty, results in the caption automatically hiding itself. The string should contain replacement values. More information about this can be accessed by clicking the 'Supported Replacement Values...' link below.

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.Caption` class in the API Reference.

DynamicText

The DynamicText element displays non-scrollable text which dynamically changes based on the currently selected gallery and image (such as the image title or caption). If the text doesn't fit in the available/allowed area, it is cropped. When the text changes, it automatically fades between the old and new values. The DynamicText element is identical to the DynamicScrollableText element except that, if the text doesn't fit in the available area, the extra text is hidden on this element whereas it can be scrolled into view on the other element.

The most important property on this element is the Displayed Text value (the `text` attribute in the XML), which controls the text displayed. Note - you can use the `htmlText` property instead of `text` if you wish to display HTML text. The Text Formatting section of the Properties Panel controls the formatting of the element's text.

About Replacement Values

The DynamicText element has support for replacement values in its label. Replacement values are special string values which are automatically replaced based on the current gallery and image's information. The following gallery-related replacement values are available:

- `{gallery.index}` or `{gallery.id}` - This string is replaced with the 1-base index of the current gallery. For example, "This is gallery #{gallery.id}" would become "This is gallery #1" while the first gallery was open, then subsequently "This is gallery #2", etc.
- `{gallery.total}` - This string is replaced with the total number of galleries. For example, "This is gallery {gallery.id}/{gallery.total}" would become "This is gallery 1/10" if there were ten galleries and the first gallery was open.
- `{gallery.index0}` or `{gallery.id0}` - This string is replaced with the **0**-base index of the current image. This property is generally only used by advanced users; for most purposes, the 1-base `{gallery.index}` value is appropriate.
- `{gallery.uniqID}` - This string is replaced with the uniqID index of the current gallery. This can be applicable when working with permalinking, and will not be affected by randomization. This property is generally only used by advanced users.

- `{gallery.rawID}` - This string is replaced with the `id` attribute defined on the actual gallery node; this is the value set in the wizard in the Gallery Linkage ID field. This is generally a property only used by advanced developers.
- `{gallery.`*`someAtt`*`}` - This string is replaced with the corresponding XML attribute of the current gallery's XML node. For example, "Current gallery: {gallery.title}" would become "Current gallery: Flowers and Sunshine" if the gallery's title was defined like this: `<gallery title="Flowers and Sunshine" ... >`. You can use the standard attributes (such as title and description), or any custom attribute which has been defined.

The following image-related replacement values are available:

- `{image.index}` or `{image.id}` - This string is replaced with the 1-base index of the current image. For example, "This is image #{image.id}" would become "This is image #1" while the first image was open, then subsequently "This is image #2", etc.
- `{image.total}` - This string is replaced with the total number of images in the current gallery. For example, "This is image {image.id}/{image.total}" would become "This is image 1/10" if there were ten images in the gallery and the first image was open.
- `{image.link}` - This string is replaced with the current image's link value (if one has been defined). For example, "Purchase from: {image.link}" would become "Purchase from: http://www.mysite.com/" if the image's link was defined like this: `<img link="http://www.mysite.com/" ... />`.
- `{image.title}` - This string is replaced with the current image's title value (if one has been defined). For example, "Current image: {image.title}" would become "Current image: Daffodil" if the image's title was defined like this: `<img ... title="Daffodil" ... />`.
- `{image.caption}` - This string is replaced with the current image's caption value (if one has been defined). For example, "About: {image.caption}" would become "About: Taken two weeks ago on holiday in South Africa" if the image's caption was defined like this: `<img ... caption="Taken two weeks ago on holiday in South Africa" ... />`.
- `{image.`*`someOtherAtt`*`}` - This string is replaced with the corresponding XML attribute of the current image's XML node. For example, "Current image: {image.title}" would become "Current image: Blue Orchid" if the image's title was defined like this: <img title="Blue Orchid" ... >. You can use the standard attributes (such as title and caption), or any custom attribute which has been defined. In other words, {image.someValue} would be replaced with `<img ... someValue="Special attribute value!" ... />`
- **[Metadata support]** `{metadata.`*`property`*`}` - This string is replaced with various available metadata parameters that are read from the loaded images. Click here to learn more about metadata integration. The available *property* values include `{metadata.creator}` (corresponds to the Creator metadata property), `{metadata.title}` (corresponds to the Title metadata property), and `{metadata.description}` (corresponds to the Description metadata property). If a requested metadata property doesn't exist on an image, or if the metadata isn't available because the image hasn't fully loaded yet, a default value can be specified using the following format: `{metadata.property`**`|Default Value Here`**`}` . After the property value, a pipe character denotes the start of the default string. For example, the following value would display *"Loading..."* until the metadata title became available: `{metadata.title|Loading...}` .

Other less-common image-related replacement values are available too:

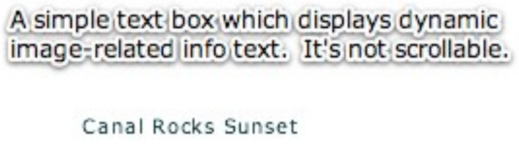
- `{image.index0}` or `{image.id0}` - This string is replaced with the **0**-base index of the current image. This is generally a property used only by advanced users; for most purposes, the 1-base `{image.index}` value is appropriate.
- `{image.uniqID}` - This string is replaced with the uniqID index of the current image. This can be used for permalinking, and will not be affected by randomization. This is generally a property only used by advanced users.

- `{image.rawID}` - This string is replaced with the id attribute defined on the actual image node; this is the value set in the wizard in the Image Linkage ID field. This is generally a property only used by advanced users.
- `{image.src}` - This string is replaced with the current image's src value. This usually represents just the file name of the image; if you want to get the entire image path, use the {image.srcFull} replacement value.
- `{image.srcFull}` - This string is replaced with the current image's full image path (use {image.src} for just the shorter path / file name).
- `{image.thumbFull}` - This string is replaced with the current image's full thumb path (use {image.thumb} for just the shorter path / file name)

For all replacement values, you can use the following format to specify a default value: `{image.property`**`|Default Value Here`**`}` . Follow the property name with a pipe character (`|`) and the default value before the closing curly brace.

Note that you can use multiple replacement values in the same string.

Here is an example of a DynamicText element in use within a slideshow:



A simple text box which displays dynamic image-related info text. It's not scrollable.

Canal Rocks Sunset

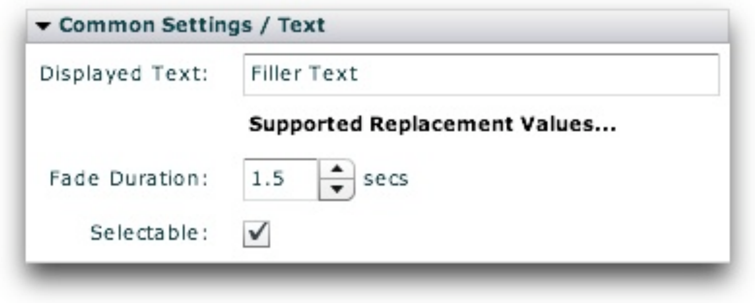
Important Usage Notes

This element is related to the DynamicLinkButton, Caption, and DynamicScrollableText elements. If you want a "pre-baked" display of the image caption and title, use the Caption element. On the other hand, this element can be used to display only one (caption or title) and customize the formatting on each separately.

If you want to display just the caption or title, you can also use the Plain Caption Text or Plain Title Text starter elements in the Special Starter Elements section.

Noteworthy Parameters

The Common Settings section of the Properties Panel contains several noteworthy parameters; they are described in the table below:



Common Settings / Text

Displayed Text: Filler Text

Supported Replacement Values...

Fade Duration: 1.5 secs

Selectable: ☑

Parameter name in Properties Panel	XML Property Name	Notes
Displayed Text	`text`	The text value displayed. Replacement values are supported. For more information about the replacement values, click the 'Supported Replacement Values...' link.
Fade Duration	`fadeTime`	Specifies the duration of the fade effect which is used to transition between different text values.
Selectable	`selectable`	Specifies if the text can be selected.

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.DynamicText` class in the API Reference.

Slideshow Control Buttons

The Slideshow Control Buttons are a set of button components which allow the user to control the slideshow and toggle various options. These buttons are often used to build control bars. When building a control bar, you may want to place the various buttons in an HBox container to automatically lay them out.

Elements in this section:

- AutoplayLoopToggleButton
- LastButton
- PlayPauseButton
- FirstButton
- PreviousButton
- LayoutJumpButton
- NextButton
- AutoplayToggleButton
- FullScreenButton
- CaptionToggleButton
- PrintButton

Advanced Users / Developers: For more detailed developer-oriented information about every element in this section, be sure to see the API reference for each element's corresponding class.

Usage Tip: When in the wizard, hover over any property name in the Properties Panel for a detailed description of what the option controls.

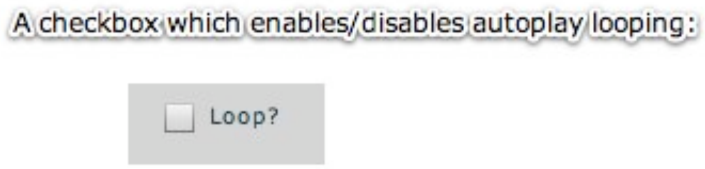
Tip: All of the elements within this section can be placed in a HoverCanvas element, which makes them only visible when the user actively moves the mouse over the slideshow. See the HoverCanvas layout container for more.

Advanced Users / Developers: Button Skinning

All of the buttons in this section can be skinned through the use of Custom Skin Files. See Custom Skin Files / Custom Buttons for more information. If you wish to only change the color of the button, change the **Theme Color** setting in the Properties Panel.

AutoplayLoopToggleButton

The AutoplayLoopToggleButton is a checkbox control which allows the user to enable or disable autoplay looping.



A checkbox which enables/disables autoplay looping:

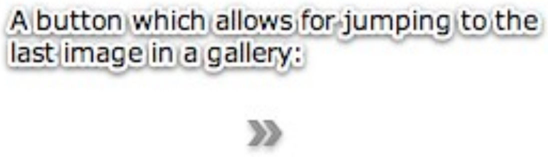
□ Loop?

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.controls.AutoplayLoopToggleButton` class in the API Reference.

LastButton

The LastButton element allows the user to jump to the final image in the current gallery. By default, the LastButton can be triggered with the keyboard shortcut of 'End'.



A button which allows for jumping to the last image in a gallery:

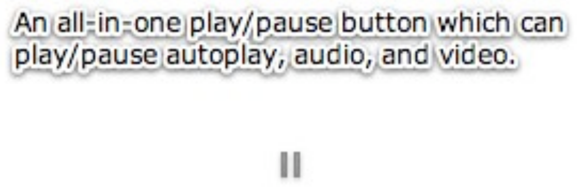
Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.controls.LastButton` class in the API Reference.

PlayPauseButton

The PlayPauseButton element allows the user to play or pause all aspects of the slideshow, including autoplay, any audio, and any video. The button can be triggered through the keyboard shortcut of 'p' (for Play/Pause) or the spacebar (may not always work properly).

The Displays Status Of and Controls Status Of properties in the Properties Panel determine which of these aspects are reflected by the button's state, and controlled when the button is clicked.

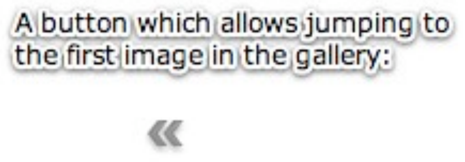

An all-in-one play/pause button which can play/pause autoplay, audio, and video.

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.controls.PlayPauseButton` class in the API Reference.

FirstButton

The FirstButton element allows the user to jump to the first image in the current gallery. By default, the FirstButton can be triggered with the keyboard shortcut of 'Home'.


A button which allows jumping to the first image in the gallery:

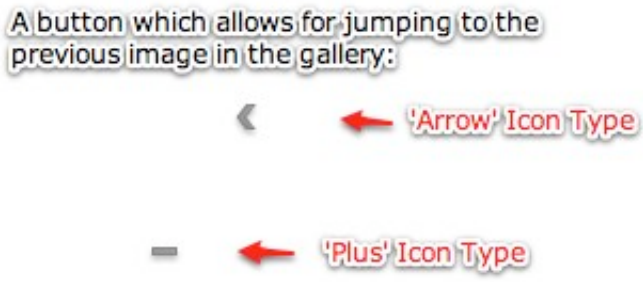
Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.controls.FirstButton` class in the API Reference.

PreviousButton

The PreviousButton element allows the user to jump to the previous image in the current gallery.   By default, the PreviousButton can be triggered with the keyboard shortcut of 'Left-arrow-key'.

You can specify which default skin is used by setting the Arrow Type parameter in the Properties Panel.



A button which allows for jumping to the previous image in the gallery:

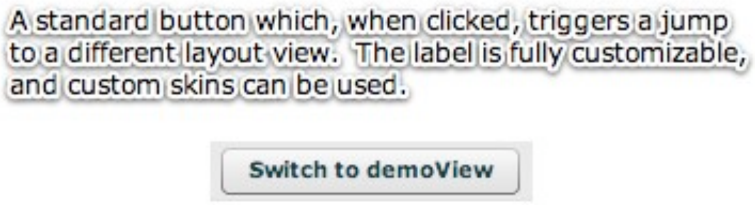
Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.controls.PreviousButton` class in the API Reference.

LayoutJumpButton

The LayoutJumpButton element, when clicked, will cause the slideshow to switch to a different layout view. This element is used to allow navigation between layout views when more than one Normal layout view exists.   For more information on layout views, see this topic.

The Destination View property in the Properties panel is the most important parameter, and is required since it specifies which view should be opened upon a click.   The 'Select a Destination View' button next to the field allows direction selection of one of the Normal layout views.



A standard button which, when clicked, triggers a jump to a different layout view. The label is fully customizable, and custom skins can be used.

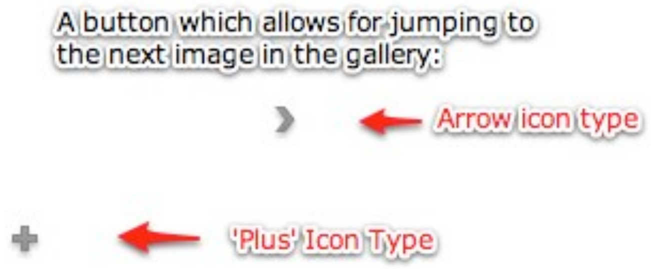
Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.LayoutJumpButton` class in the API Reference.

NextButton

The NextButton element allows the user to jump to the next image in the current gallery.   By default, the NextButton can be triggered with the keyboard shortcut of 'Right-arrow-key'.

You can specify which default skin is used by setting the Arrow Type parameter in the Properties Panel.



A button which allows for jumping to the next image in the gallery:

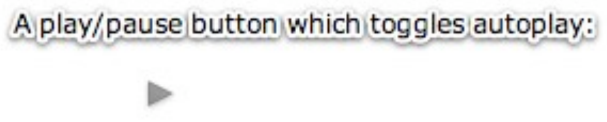
Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.controls.NextButton` class in the API Reference.

AutoplayToggleButton

The AutoplayToggleButton element allows the user to enable or disable autoplay. The button can be triggered through the keyboard shortcut of 'p' (for Play/Pause) or the spacebar (may not always work properly).

If you want to control both autoplay and audio or video all at the same time, use the PlayPauseButton element instead.



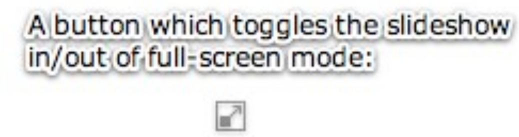
A play/pause button which toggles autoplay:

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.controls.AutoplayToggleButton` class in the API Reference.

FullScreenButton

The FullScreenButton element allows the user to toggle in and out of full-screen mode. By default, the FullScreenButton can be triggered with the keyboard shortcut of 'f' (for Full-screen).



A button which toggles the slideshow in/out of full-screen mode:

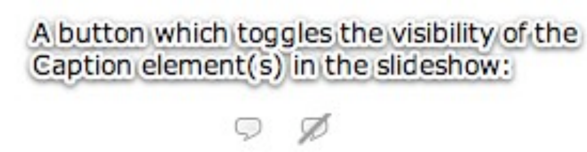
Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.controls.FullScreenButton` class in the API Reference.

CaptionToggleButton

The CaptionToggleButton element displays the default caption show/hide icons; when clicked, it will by default show/hide any Caption element(s) in the slideshow.   The default handler can be disabled, allowing for it to trigger any other behavior as well.

By default, the CaptionToggleButton can be triggered with the keyboard shortcut of 'i' (for Information).



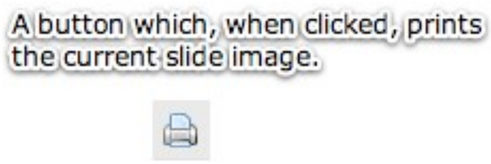
A button which toggles the visibility of the Caption element(s) in the slideshow:

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.controls.CaptionToggleButton` class in the API Reference.

PrintButton

The PrintButton element displays the default printing icon; when clicked, it will by default print the current slide's image. The default handler can be disabled, allowing for it to trigger any other behavior as well. The button is automatically disabled if there's no slide to print.



A button which, when clicked, prints the current slide image.

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.PrintButton` class in the API Reference.

Slideshow Option Controls

The Slideshow Option Controls include several elements which are occasionally used to give control over various options.

Elements in this section:

- AutoplayDelaySlider
- GotoBox
- PagingControlSizer
- AutoplayProgressDisplay

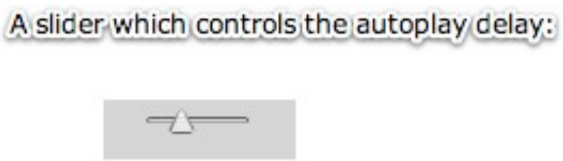
Advanced Users / Developers: For more detailed developer-oriented information about every element in this section, be sure to see the API reference for each element's corresponding class.

Usage Tip: When in the wizard, hover over any property name in the Properties Panel for a detailed description of what the option controls.

Tip: All of the elements within this section can be placed in a HoverCanvas element, which makes them only visible when the user actively moves the mouse over the slideshow. See the HoverCanvas layout container for more.

AutoplayDelaySlider

The AutoplayDelaySlider is a slider control which allows the user to adjust the autoplay delay between images. The Minimum and Maximum Delay parameters in the Properties Panel allow you to specify the possible delay range.



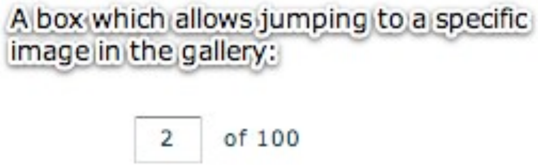
A slider which controls the autoplay delay:

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.controls.AutoplayDelaySlider` class in the API Reference.

GotoBox

The GotoBox control element allows the user to type in the cardinal index of an image/slide and press 'enter' to jump directly to it. The control is made up of a TextInput box which displays the current image index, and a Label which displays the total number of images.



A box which allows jumping to a specific image in the gallery:

Equivalement Class File

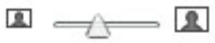
For detailed reference on this element, see the `com.dwuser.ss4.ui.controls.GotoBox` class in the API Reference.

PagingControlSizer

Note: *Implementing this element requires code editing. Only use it if you are comfortable with editing the XML code.*

The PagingControlSizer element is a control which allows the user to resize the tiles in any element which extends PagingControlHolder, specifically the NumberBar and Thumbnails elements. It is often used in conjunction with the `<PagingControls>...</PagingControls>` child descriptor of the NumberBar or Thumbnails element as shown in the example in the class reference. For more on this use, see the `PagingControls` pseudo-property on the

PagingControlHolder class. However, it can be used independently as long as the pagingOwner property has been set. For a code sample / example of this type of implementation, see the class reference.



A slider which allows the user to change the size of thumbnails or tiles in a Thumbnails or NumberBar element:

Equivalement Class File

For detailed reference on this element, see the com.dwuser.ss4.ui.controls.PagingControlSizer class in the API Reference.   As noted above, also see the com.dwuser.ss4.ui.PagingControlHolder class (the parent class of the Thumbnails and NumberBar elements).

AutoplayProgressDisplay

The AutoplayProgressDisplay element displays a small circle indicator which shows the autoplay timer progress. This allows the user to know how long until the slideshow automatically progresses to the next image.

The ring fills as the autoplay timer progresses, indicating how long until the next image appears:

Equivalement Class File

For detailed reference on this element, see the com.dwuser.ss4.ui.controls.AutoplayProgressDisplay class in the API Reference.

Extras / Decorations

The Extras / Decorations include several elements which can be used to enhance or tweak the visual appearance of a slideshow.

Elements in this section:

- HRule
- VRule
- BorderBox
- Spacer

Advanced Users / Developers: For more detailed developer-oriented information about every element in this section, be sure to see the API reference for each element's corresponding class.

Usage Tip: When in the wizard, hover over any property name in the Properties Panel for a detailed description of what the option controls.

HRule

This element is a horizontal separator line which can be used to provide visual delineation between separate parts of a layout. For instance, it might be used between two children in a VBox container.

By default, it has a width of `100%`. This can be changed as needed.

Equivalement Class File

For detailed reference on this element, see the `mx.controls.HRule` class in the API Reference.

VRule

This element is a vertical separator line which can be used to provide visual delineation between separate parts of a layout. For instance, it might be used between two children in an HBox container.

By default, it has a height of `100%`. This can be changed as needed.

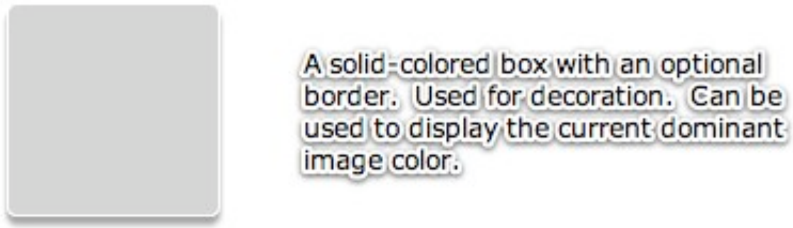
Equivalement Class File

For detailed reference on this element, see the `mx.controls.VRule` class in the API Reference.

BorderBox

The BorderBox element displays a solid background / border. When its `backgroundColor` is changed, the transition is made with a smooth fade transition.

The element is designed to be used as a decoration which only displays a border, background, and/or drop shadow.



A solid-colored box with an optional border. Used for decoration. Can be used to display the current dominant image color.

The element can also be used to display the current image's dominant color. For more information, see the Dominant-slide-color BorderBox special starter element.

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.extras.BorderBox` class in the API Reference.

Spacer

This element makes a spacer appear between child elements in a box-layout container such as HBox or VBox. It is essentially an empty box which can be used to adjust the position of other elements.

For example, you might have an HBox with a width of `100%` and add a PreviousButton and NextButton to it. If you wanted to force the PreviousButton to be on the left edge and the NextButton on the right edge, you could add a Spacer element between the two buttons and set its width to `100%` as well. This would make it fill the entire box, forcing the NextButton to the far right edge.

Equivalement Class File

For detailed reference on this element, see the `mx.controls.Spacer` class in the API Reference.

Audio Control Elements

Audio Control Elements are a group of elements which can be used to add audio controls. This allows the user to control any audio which has been added to the slideshow.

Elements in this section:

- AudioPlayPauseButton*
- AudioControlCanvas
- AudioScrubBar*
- AudioMuteButton
- AudioControlBar *(a simple audio control bar -- good for beginners)*
- AudioLoudButton
- AudioVolumeSlider
- AudioVolumeToggleButton
- AudioVolumeControl
- AudioLoadBar*
- AudioInfoText*

***Important Note:** It is important to note that the individual audio control elements with an asterisk should be placed within the AudioControlCanvas element. The reference below for this item should be reviewed first.

Advanced Users / Developers: For more detailed developer-oriented information about every element in this section, be sure to see the API reference for each element's corresponding class.

Usage Tip: When in the wizard, hover over any property name in the Properties Panel for a detailed description of what the option controls.

Advanced Users / Developers: Button Skinning

All of the buttons in this section can be skinned through the use of Custom Skin Files. See Custom Skin Files / Custom Buttons for more information.   If you wish to only change the color of the button, change the **Theme Color** setting in the Properties Panel.

AudioPlayPauseButton

The AudioPlayPauseButton plays or pauses the currently playing audio.   **Important:** To function properly, this element should be within an AudioControlCanvas or AudioControlBar. This tells it which level of audio to control (intro/background, gallery, and/or image).

Note that this element only controls audio; it doesn't affect Autoplay.   Use the PlayPauseButton element if you wish to play/pause everything (audio, video, autoplay).

The color of the button can be customized through the Theme Color setting in the Properties Panel.

Example element in use within a slideshow:

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.audiocontrols.AudioPlayPauseButton` class in the API Reference.

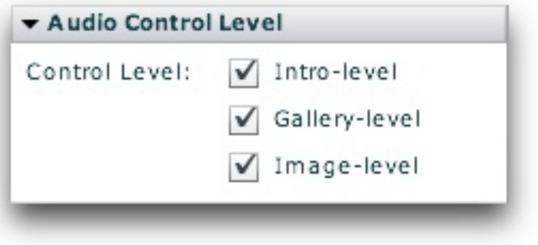
AudioControlBar

The AudioControlBar class provides a default audio control component when a simple audio controller is needed. It incorporates both an AudioVolumeToggleButton and an AudioInfoText element, and has a nice fade effect when showing/hiding.   The element can also be used to hold other custom audio control elements for an audio controller.   It is special because, like its parent AudioControlCanvas, it automatically assigns the proper level setting to all child elements based on its own Control Level setting.   This is important because it ensures that only the applicable type(s) of audio are controlled by a given controller.

This element, as a Canvas and subclass of AudioControlCanvas, is usually best used by adding a layout container (such as an HBox) which holds the actual audio control elements. The AudioControlCanvas (or it's child containers) can contain the following audio-control child elements:

AudioInfoText, AudioLoadBar, AudioLoudButton, AudioMuteButton, AudioPlayPauseButton, AudioScrubBar, AudioVolumeControl, AudioVolumeSlider, AudioVolumeToggleButton.

Important! Be sure to review the value for the Control Level attribute (pictured below). If you don't specify the correct and applicable level value, the audio controller **won't be visible when it should be**. The default value means that this controller will be visible all types of audio is playing. If you want only certain audio to be controlled, select only some boxes.



Control Level: ☑ Intro-level
☑ Gallery-level
☑ Image-level

As a review: Intro-level audio is the slideshow background audio (specified in the Global Configuration & Playback Settings section); Gallery-level audio is audio associated with a single specific gallery; Image-level audio is audio associated with a single specific image.

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.audiocontrols.AudioControlBar` class in the API Reference.

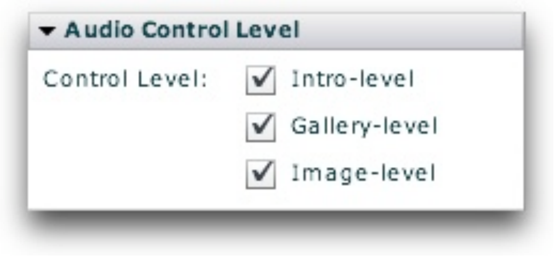
AudioControlCanvas

The AudioControlCanvas element is used to hold all audio control elements for an audio controller. It is special because it automatically assigns the proper level setting to all child elements based on its own Control Level setting. This is important because it ensures that only the applicable type(s) of audio are controlled by a given controller.

This element, as a Canvas, is usually best used by adding a layout container (such as an HBox) which holds the actual audio control elements. The AudioControlCanvas (or it's child containers) can contain the following audio-control child elements:

AudioInfoText, AudioLoadBar, AudioLoudButton, AudioMuteButton, AudioPlayPauseButton, AudioScrubBar, AudioVolumeControl, AudioVolumeSlider, AudioVolumeToggleButton.

Important! Be sure to review the value for the Control Level attribute (pictured below). If you don't specify the correct and applicable level value, the audio controller **won't be visible when it should be**. The default value means that this controller will be visible all types of audio is playing. If you want only certain audio to be controlled, select only some boxes.



Audio Control Level

Control Level: ☑ Intro-level
☑ Gallery-level
☑ Image-level

As a review: Intro-level audio is the slideshow background audio (specified in the Global Configuration & Playback Settings section); Gallery-level audio is audio associated with a single specific gallery; Image-level audio is audio associated with a single specific image.

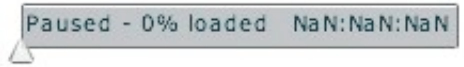
Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.audiocontrols.AudioControlCanvas` class in the API Reference.

AudioScrubBar

The AudioScrubBar element allows the user to scrub to a specific point in the currently playing audio, while displaying the current progress. **Important:** To function properly, this element should be within an AudioControlCanvas or AudioControlBar. This tells it which level of audio to control (intro/background, gallery, and/or image).

The appearance / theme color of the element can be customized through the Theme Color setting in the Properties Panel.



Paused - 0% loaded NaN:NaN:NaN

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.audiocontrols.AudioPlayPauseButton` class in the API Reference.

AudioMuteButton

The AudioMuteButton element is a button which, when clicked, mutes the audio volume. This element will control the volume of all audio on the slideshow; it is a "global" volume setting. If you would rather have a toggled button which functions as both a "loud" and "soft" button, use the AudioVolumeToggleButton element.

The color of the button can be customized through the Theme Color setting in the Properties Panel.

Example:

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.audiocontrols.AudioMuteButton` class in the API Reference.

AudioLoudButton

The AudioLoudButton element is a button which, when clicked, sets the volume to full strength. This element will control the volume of all audio on the slideshow; it is a "global" volume setting. If you would rather have a toggled button which functions as both a "loud" and "soft" button, use the AudioVolumeToggleButton element.

The color of the button can be customized through the Theme Color setting in the Properties Panel.

Example:

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.audiocontrols.AudioLoudButton` class in the API Reference.

AudioVolumeSlider

The AudioVolumeSlider element allows the user to control the audio volume level by dragging a slider. This element will control the volume of all audio on the slideshow; it is a "global" volume setting.

The appearance / theme color of the element can be customized through the Theme Color setting in the Properties Panel.

Example:

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.audiocontrols.AudioVolumeSlider` class in the API Reference.

AudioVolumeToggleButton

The AudioVolumeToggleButton element allows the user to toggle the volume back and forth from full-level to muted. This element will control the volume of all audio on the slideshow; it is a "global" volume setting.

In the Properties Panel, you can set the initial state as well as the full volume level. As with all components, hover over the settings in the Properties Panel for a detailed description of that parameter.

The color of the button can be customized through the Theme Color setting in the Properties Panel.

Example (in full-level state):

Equivalement Class File

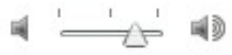
For detailed reference on this element, see the `com.dwuser.ss4.ui.audiocontrols.AudioVolumeToggleButton` class in the API Reference.

AudioVolumeControl

The AudioVolumeControl element allows the user to control the audio volume, incorporating three separate elements: AudioMuteButton, AudioVolumeSlider, and AudioLoudButton.   There is a "mute" button, a volume slider, and a "loud" button. This element will control the volume of all audio on the slideshow; it is a "global" volume setting.

The appearance / theme color of the element can be customized through the Theme Color setting in the Properties Panel.

Example:



Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.audiocontrols.AudioVolumeControl` class in the API Reference.

AudioLoadBar

The AudioLoadBar element shows a loading-progress bar for the currently streaming audio. A solid track is drawn using the specified background color, then it is overlaid with the fill color as the audio loads. When the loading is fully complete, the fill color fully overlays the background color. **Important:** To function properly, this element should be within an AudioControlCanvas or AudioControlBar. This tells it which level of audio to control (intro/background, gallery, and/or image).

The Background Color and Fill Color parameters in the Properties Panel control the appearance of the bar.

Example:

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.audiocontrols.AudioLoadBar` class in the API Reference.

AudioInfoText

The AudioInfoText element displays information about the currently playing audio. **Important:** To function properly, this element should be within an AudioControlCanvas or AudioControlBar. This tells it which level of audio to control (intro/background, gallery, and/or image).

The Label Text property in the Properties Panel controls the text displayed. There are a set of special replacement values which can be included in the label to specify information relating to the currently playing audio file:

- `{status}` - Displays the current status, either Playing or Paused. The specific text displayed for each state depends on the playingLabel and pausedLabel style parameters.
- `{label}` - Displays the label associated with this audio file. This is the Audio Label field in the wizard, or the audioLabel attribute in code view.
- `{percent_loaded}` - Displays the percent of the audio file which has been loaded.
- `{totaltime}` - Displays the total duration of the audio. While the file is still loading, this value can change as the loading takes place (because the actual value is not known until loading completes). To avoid any confusion to your users, you can specify a certain value to be displayed in place of the

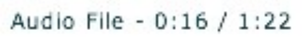
duration while the file is still loading. This is done through the loadingDurationLabel parameter.

- `{time}` - Displays the current position / playhead time within the audio file.

The default label value is `{label} - {time} / {totaltime}` which becomes, for example, "Audio File - 0:16 / 1:22" after 16 seconds of a 1:22 audio file labeled 'Audio File'.

The appearance of the text can be controlled through the Text Formatting section of the Properties Panel.

Example:



Audio File - 0:16 / 1:22

Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.audiocontrols.AudioInfoText` class in the API Reference.

Static Generic Elements

The Static Generic Elements are a group of non-changing generic elements which are helpful for building a slideshow user interface.

Elements in this section:

- LinkButton
- FlexVideoPlayer
- Image
- Label
- Button
- Text

Advanced Users / Developers: For more detailed developer-oriented information about every element in this section, be sure to see the API reference for each element's corresponding class.

Usage Tip: When in the wizard, hover over any property name in the Properties Panel for a detailed description of what the option controls.

Advanced Users / Developers: Button Skinning

All of the buttons in this section can be skinned through the use of Custom Skin Files. See Custom Skin Files / Custom Buttons for more information. If you wish to only change the color of the button, change the **Theme Color** setting in the Properties Panel.

LinkButton

The LinkButton is a generic button with text and a rollover / select effect. It can have a custom click handler attached to it and be used to redirect to a different page or perform some other action. For example, you might use it to create a 'Site Homepage' button which redirects the user back to a different page.

The Label setting in the Property Inspector controls the button's text. The Click Handler setting controls what happens when the button is clicked; the handler will be Actionscript code. The default handler code is for a URL redirect (the URL just needs to be added).

Example element in use within a slideshow:

Equivalement Class File

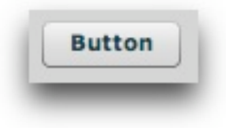
For detailed reference on this element, see the `mx.controls.LinkButton` class in the API Reference.

Button

The Button element creates a simple button which can be used to control other settings or values. It can have a custom click handler attached to it and be used to redirect to a different page or perform some other action. For example, you might use it to toggle the visibility of some other element.

The Label setting in the Property Inspector controls the button's text. The Click Handler setting controls what happens when the button is clicked; the handler will be Actionscript code. The default handler code is for a URL redirect (the URL just needs to be added).

Example element in use within a slideshow:



Equivalement Class File

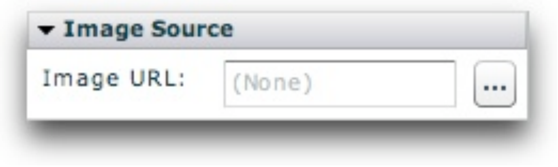
For detailed reference on this element, see the `mx.controls.Button` class in the API Reference.

Image

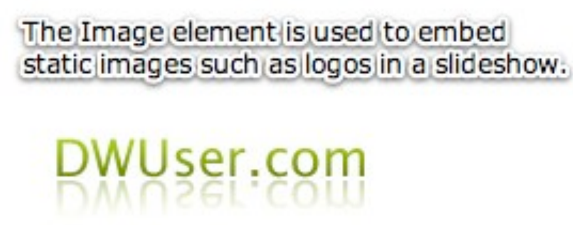
The Image element allows for a non-changing image or SWF to be displayed in the slideshow. For example, it can be a background image or a logo.

The URL of the video file (which can be a JPG, PNG, SWF, or GIF) is specified in the Image URL field in the Properties Panel. The URL must be relative to the

HTML page holding the slideshow, and may be affected by loading security restrictions if loading from a different domain. The Browse button ('...') can be used to select a file directly:



Example element in use within a slideshow:


The Image element is used to embed static images such as logos in a slideshow.

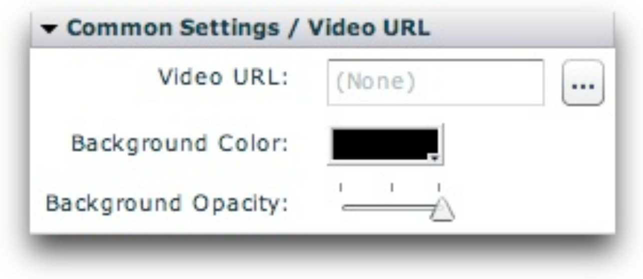
Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.ui.Image` class in the API Reference.

FlexVideoPlayer

The FlexVideoPlayer element allows for a separate video (which is not associated with any image) to be displayed within the slideshow.   This element should be rarely used. Note that it is NOT needed and totally unrelated for playing video associated with a specific slide.

The URL of the video file (which can be an FLV, MPEG4 Hi-Def (H.264), or an RTMP streaming URL) is specified in the Video URL field in the Properties Panel. The URL must be relative to the HTML page holding the slideshow, and may be affected by loading security restrictions if loading from a different domain. The Browse button ('...') can be used to select a file directly:



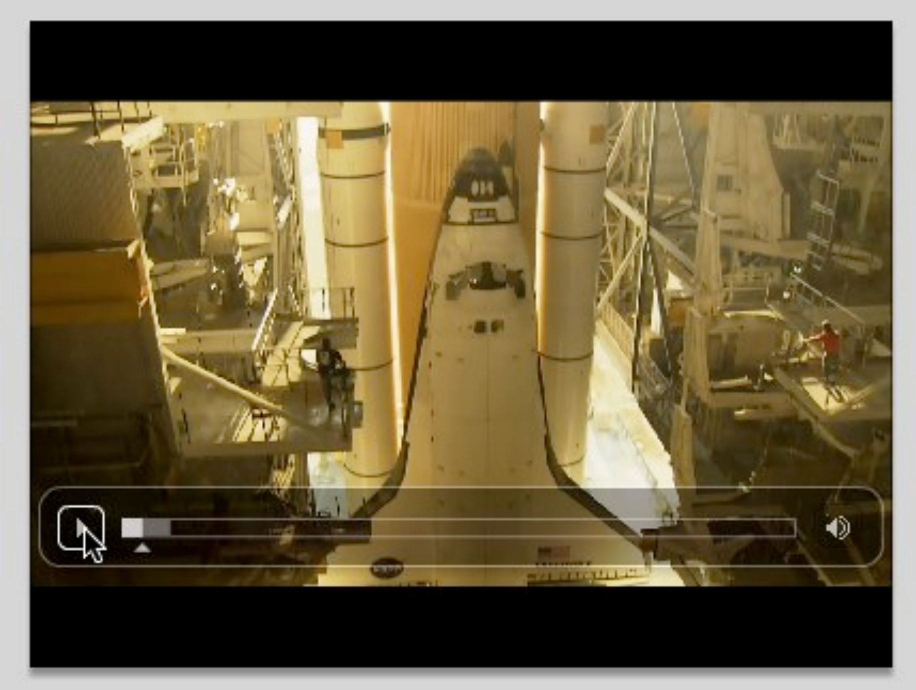
Common Settings / Video URL

Video URL: (None)

Background Color:

Background Opacity:

Example element in use within a slideshow:



Equivalement Class File

For detailed reference on this element, see the `com.dwuser.ss4.video.FlexVideoPlayer` class in the API Reference.

Label

The Label element displays a single-line label which contains non-changing text. For example, you might use it for a slideshow title (e.g. 'My Portfolio').

The Displayed Text setting in the Property Inspector controls the button's text. You can manually edit the `htmlText` property if you want to include HTML; if editing the XML directly, you must encode all HTML.

Example element in use within a slideshow:

Filler text

Equivalement Class File

For detailed reference on this element, see the `mx.controls.Label` class in the API Reference.

Text

The Text element displays a label which contains non-changing text. It differs from the Label element in that it supports multiple lines of text. For example, you might use it for an "about-me" description paragraph.

The Displayed Text setting in the Property Inspector controls the button's text. You can manually edit the `htmlText` property if you want to include HTML; if editing the XML directly, you must encode all HTML.

Example element in use within a slideshow:

Filler text

Equivalement Class File

For detailed reference on this element, see the `mx.controls.Text` class in the API Reference.

Using the External Wizard to Create and Edit Slideshows > Section-by-Section Usage Guide > Using the 'Layout / Presentation' Section > Available Elements (incl. Cheatsheet) > **Special Starter Elements**

Special Starter Elements

The Special Starter Elements are pre-baked elements which can be used to save time. The incorporate other elements in one neat starter package; after being added, their constituent components can be edited individually.

Elements in this section:

- Dominant-slide-color BorderBox
- Plain Title Text
- Plain Caption Text
- Centered Loaders Block
- Pre-Built Hover-only Control Bar - NumberBar
- Pre-Built Hover-only Control Bar - Thumbnails
- Pre-Built Control Bar - NumberBar
- Pre-Built Control Bar - Thumbnails

Dominant-slide-color BorderBox

This special starter element is a BorderBox whose color automatically changes to match the dominant color of the current image. It makes your slideshow have a background color which always "matches". You'll usually want to put this element in an underlay layer. Inspect the element in code view to learn about the special attribute binding values necessary to accomplish this.

This is a special kind of BorderBox; here is an example of a generic BorderBox:

A solid-colored box with an optional border. Used for decoration. Can be used to display the current dominant image color.

Note: All the individual elements which comprise a starter element can be individually edited after the starter element has been added.

Equivalement Class File

This is a compound starter element made up of other elements; it has no individual associated class file.

The Layout XML for the starter element is as follows:

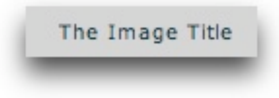
```
<BorderBox width="150" height="150"
backgroundColor="Bind{slideshow.currentPreSyncSlide.dominantColorLight}" borderThickness="0"
borderStyle="solid" backgroundChangeEffect="true" backgroundChangeEffectDuration="1700" />
```

Plain Title Text

This special starter element is a plain text box which displays the current image title. If you just want the plain text displayed (without any border or background), add this element. To learn more about the element code, inspect the element in code view.

The formatting of the text can be controlled through the Text Formatting section of the Properties Panel.

Example:



The Image Title

Noteworthy Parameters

See the DynamicScrollableText element for more information about specific parameters.

Equivalement Class File

This is a compound starter element made up of other elements; it has no individual associated class file.

The Layout XML for the starter element is as follows:

```
<DynamicScrollableText htmlText="{image.title}" />
```

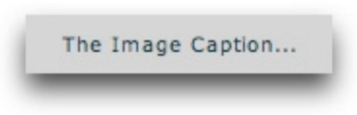
Plain Caption Text

This special starter element is a plain text box which displays the current image caption / description. If you just want the plain text displayed (without any border or background), add this element. To learn more about the element code, inspect the element in code view.

By default, the element has a maximum height of 200 pixels; if, based on the specified width, the height is greater than that, a scrollbar will appear.

The formatting of the text can be controlled through the Text Formatting section of the Properties Panel.

Example:



Noteworthy Parameters

See the DynamicScrollableText element for more information about specific parameters.

Equivalement Class File

This is a compound starter element made up of other elements; it has no individual associated class file.

The Layout XML for the starter element is as follows:

```
<DynamicScrollableText htmlText="{image.caption}" maxHeight="200" />
```

Centered Loaders Block

This element is a duplicate of the Centered Loaders Block special starter element in the Image Progress Loaders section. See that topic for information on the element.

Pre-Built Hover-only Control Bar - NumberBar

This element is a duplicate of the *Pre-Built Hover-only Control Bar - NumberBar* special starter element in the Main Slideshow Elements section. See that topic for information on the element.

Pre-Built Hover-only Control Bar - Thumbnails

This element is a duplicate of the *Pre-Built Hover-only Control Bar - Thumbnails* special starter element in the Main Slideshow Elements section. See that topic for information on the element.

Pre-Built Control Bar - NumberBar

This element is a duplicate of the *Pre-Built Control Bar - NumberBar* special starter element in the Main Slideshow Elements section. See that topic for information on the element.

Pre-Built Control Bar - Thumbnails

This element is a duplicate of the *Pre-Built Control Bar - Thumbnails* special starter element in the Main Slideshow Elements section. See that topic for information on the element.

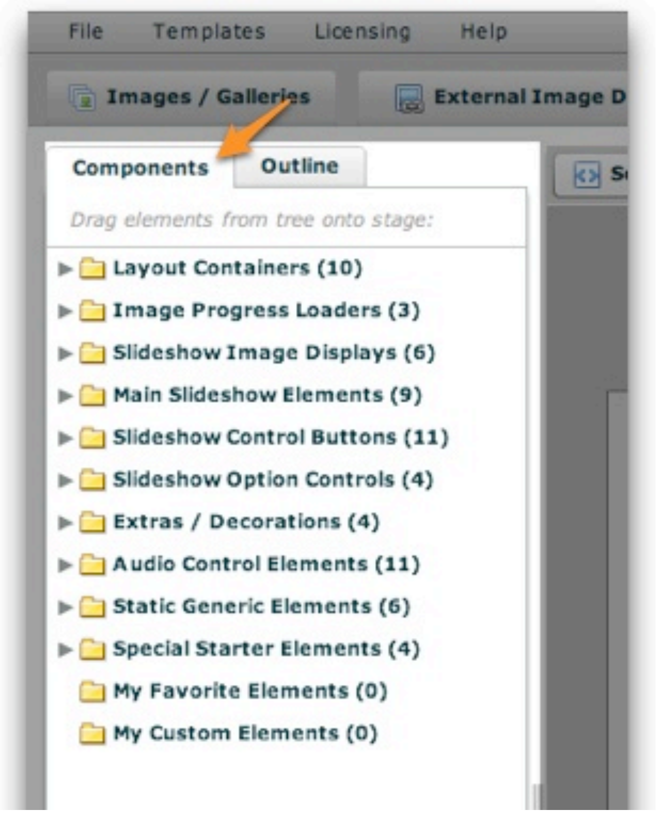
Using the External Wizard to Create and Edit Slideshows > Section-by-Section Usage Guide > Using the 'Layout / Presentation' Section > **Using the Design Area and Components Panel**

Using the Design Area and Components Panel

Armed with a knowledge of all the available elements, you're ready to start fleshing out the various layout views. The Design Area and Components Panel are the home-base for doing this.

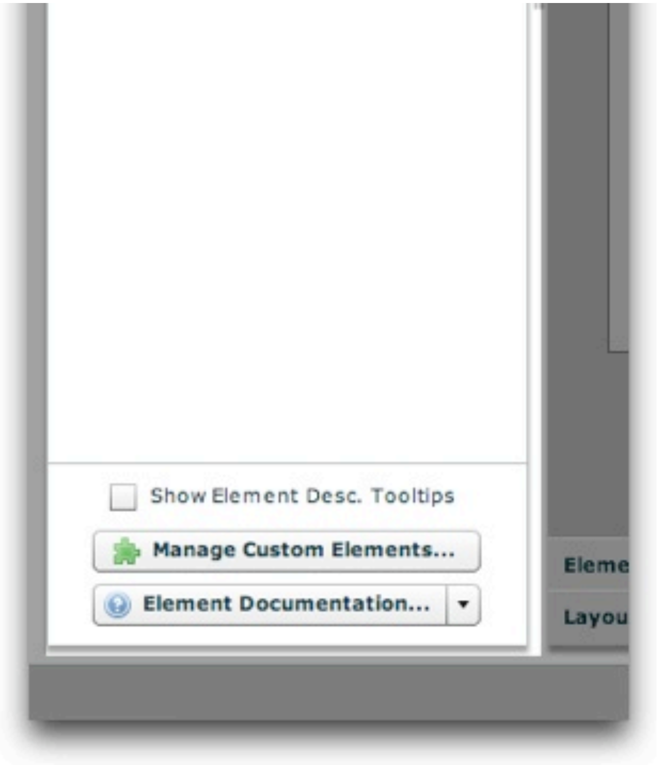
The Components Panel

The Components Panel holds all of the available elements. The panel, located on the left side of the Layout / Presentation section, appears as follows:

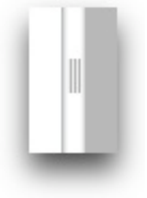


Drag elements from tree onto stage:

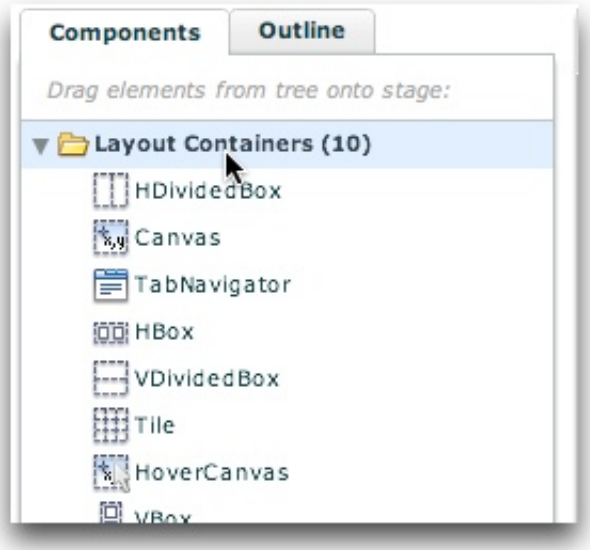
- Layout Containers (10)
- Image Progress Loaders (3)
- Slideshow Image Displays (6)
- Main Slideshow Elements (9)
- Slideshow Control Buttons (11)
- Slideshow Option Controls (4)
- Extras / Decorations (4)
- Audio Control Elements (11)
- Static Generic Elements (6)
- Special Starter Elements (4)
- My Favorite Elements (0)
- My Custom Elements (0)



To resize the panel, use the draggable resizer on the right edge of the box:

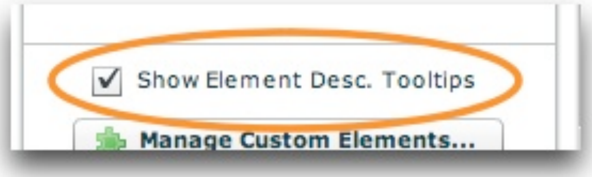


There are a number of folders, each representing a different type of elements. All of the elements are described in detail in the Available Elements section. The Favorite elements and Custom Elements folders are described below in more detail. To expand a certain type of element, click on its header:



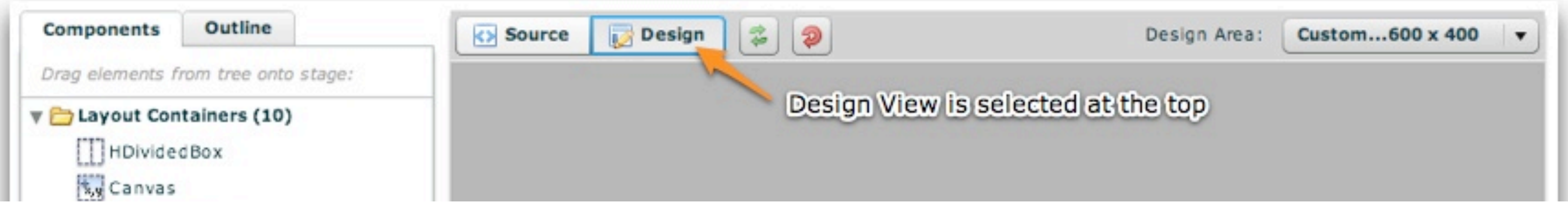
When you close and later re-open the panel, the folders will be expanded or collapsed as appropriate to restore their previous state.

If you want a quick description of each element which in the wizard, enable the 'Show Element Description Tooltips' option at the bottom:

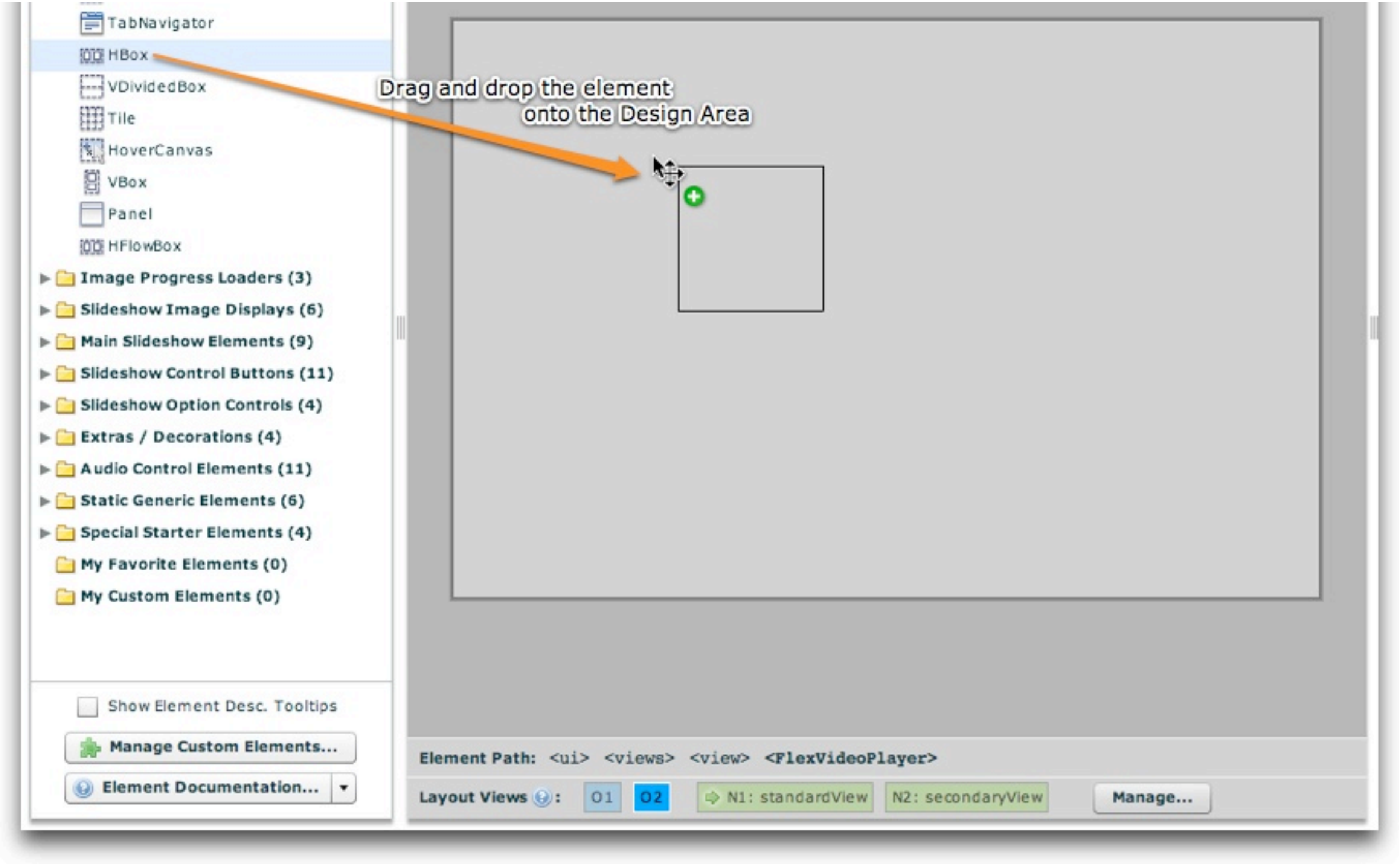


Then, a description tooltip will be shown when you hover over an item in the tree.

To add an element in the Components Panel in your slideshow, simply drag it onto the Design Area (see below for more information using the Design Area):



Design View is selected at the top



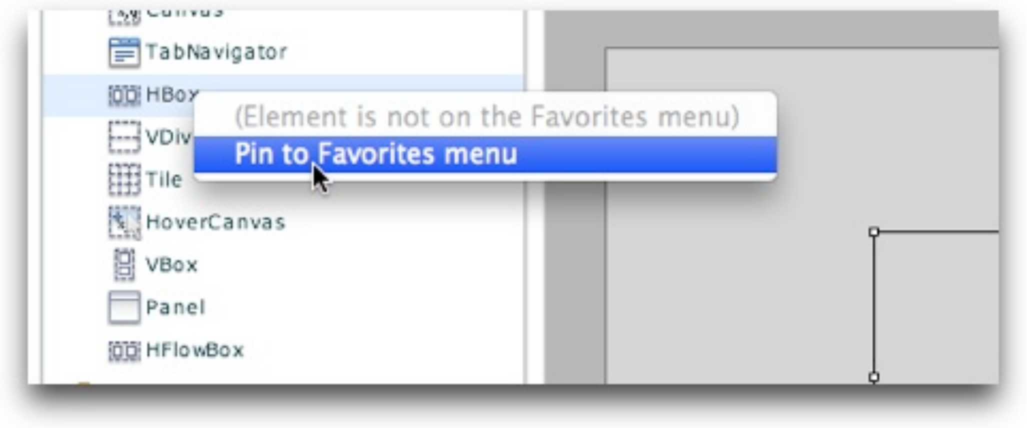
TabNavigator
HBox
VDividedBox
Tile
HoverCanvas
VBox
Panel
HFlowBox
Image Progress Loaders (3)
Slideshow Image Displays (6)
Main Slideshow Elements (9)
Slideshow Control Buttons (11)
Slideshow Option Controls (4)
Extras / Decorations (4)
Audio Control Elements (11)
Static Generic Elements (6)
Special Starter Elements (4)
My Favorite Elements (0)
My Custom Elements (0)
□ Show Element Desc. Tooltips
Manage Custom Elements...
Element Documentation...
Layout Views
O1
O2
N1: standardView
N2: secondaryView
Manage...

Drag and drop the element onto the Design Area

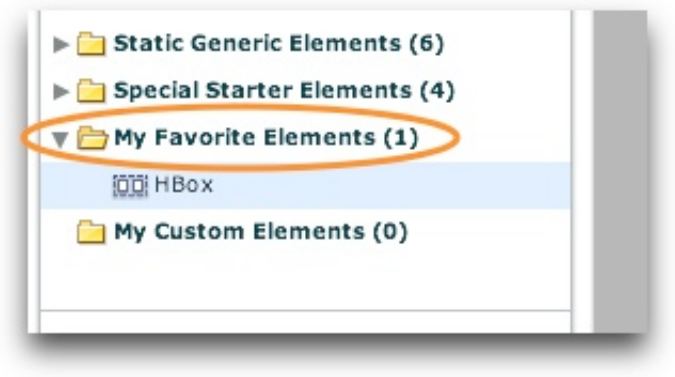
Element Path: <ui> <views> <view> <FlexVideoPlayer>

The Components Panel: Using Favorite Elements

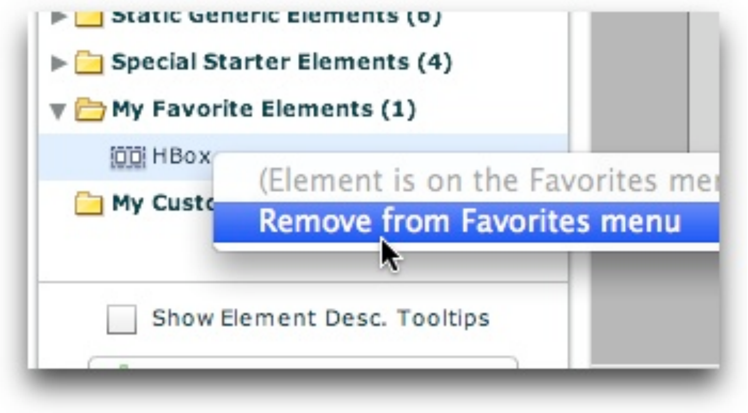
Instead of navigating from section to section in the Components Panel, you can pin selected elements to the My Favorite Elements menu. This allows one-click access to your most commonly used elements. To designate an element as a favorite, right-click (Cmd+click) it and select 'Pin to Favorites Menu':



This will add it to the favorites section, which is accessed through the My Favorite Elements folder:

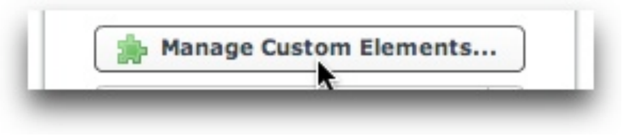


To remove an item from the favorites menu, right-click on it (Cmd+click) and select 'Remove from Favorites Menu':

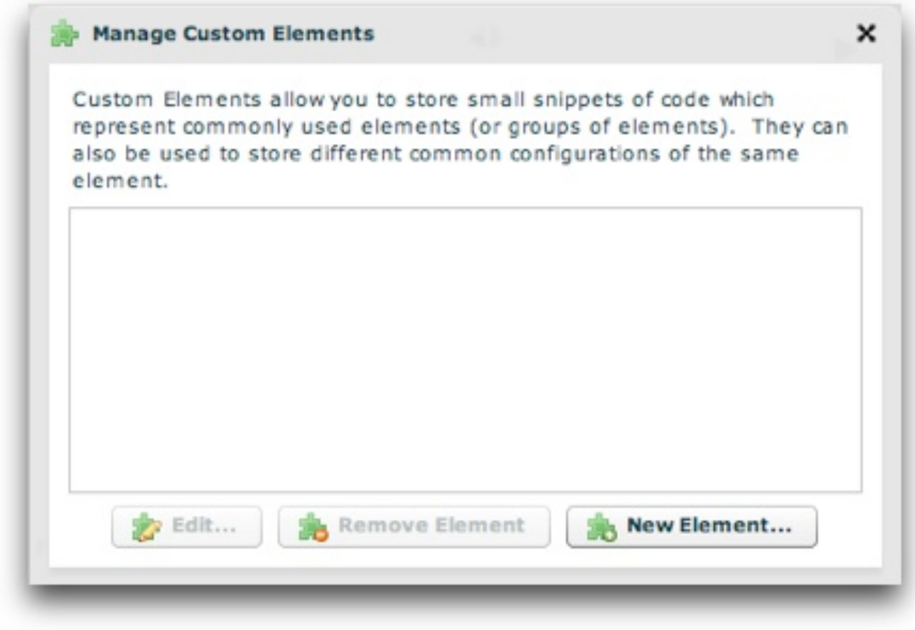

Static Generic Elements (6)
Special Starter Elements (4)
My Favorite Elements (1)
HBox
My Custo
(Element is on the Favorites me
Remove from Favorites menu
Show Element Desc. Tooltips

The Components Panel: Using Custom Elements

Custom Elements allow you to store small snippets of code which represent commonly used elements (or groups of elements). They can also be used to store different common configurations of the same element. To get started using custom elements, click the Manage Custom Elements button at the bottom of the Components Panel:

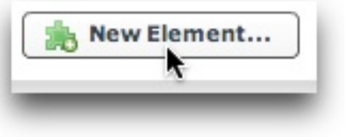


This will open the Custom Element Management dialog:

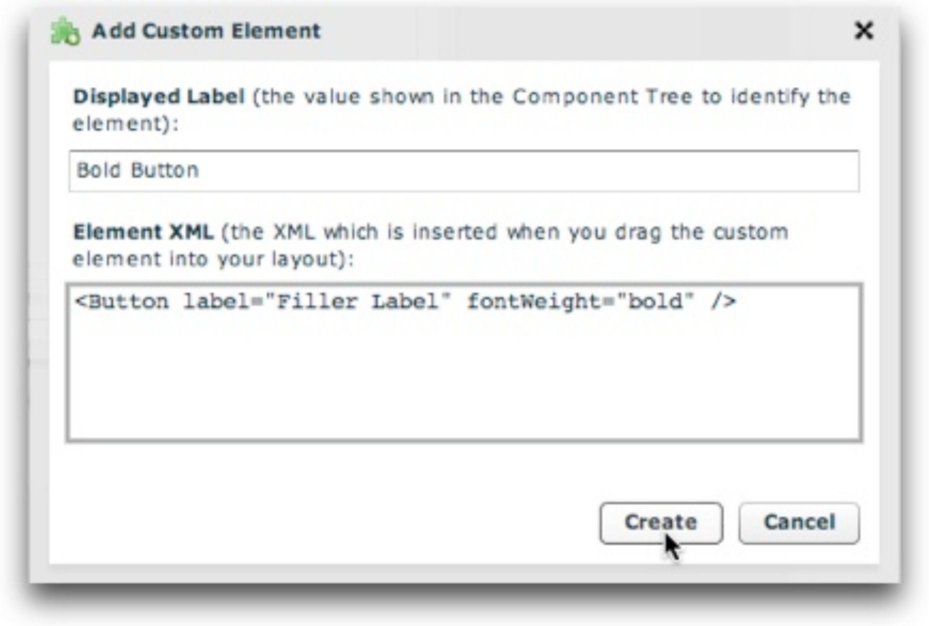


Custom Elements allow you to store small snippets of code which represent commonly used elements (or groups of elements). They can also be used to store different common configurations of the same element.

To begin, click the New Element... button:



This launches the New Custom Element dialog:



Add Custom Element

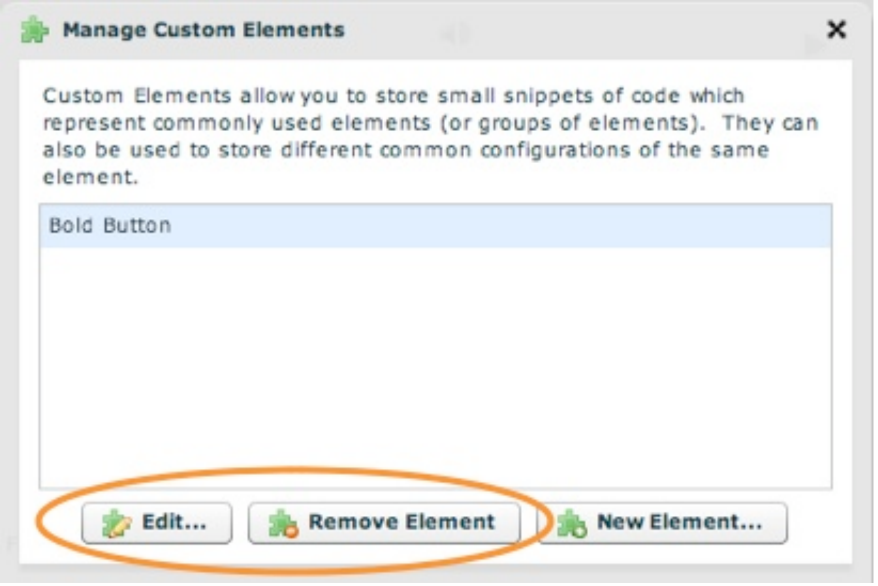
Displayed Label (the value shown in the Component Tree to identify the element):

Bold Button

Element XML (the XML which is inserted when you drag the custom element into your layout):

```
<Button label="Filler Label" fontWeight="bold" />
```

Enter the Displayed Label (the value shown in the Component Tree to identify the element) and the Element XML (the XML which is inserted when you drag the custom element into your layout). Click the Create button to add the new element. It will created and appear in the Custom Element Management dialog. If you want to edit or remove an existing custom element, select it in the grid and click 'Edit' or 'Remove Element' at the bottom:

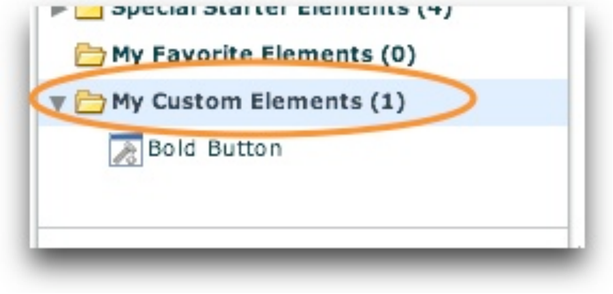


Manage Custom Elements

Custom Elements allow you to store small snippets of code which represent commonly used elements (or groups of elements). They can also be used to store different common configurations of the same element.

Bold Button

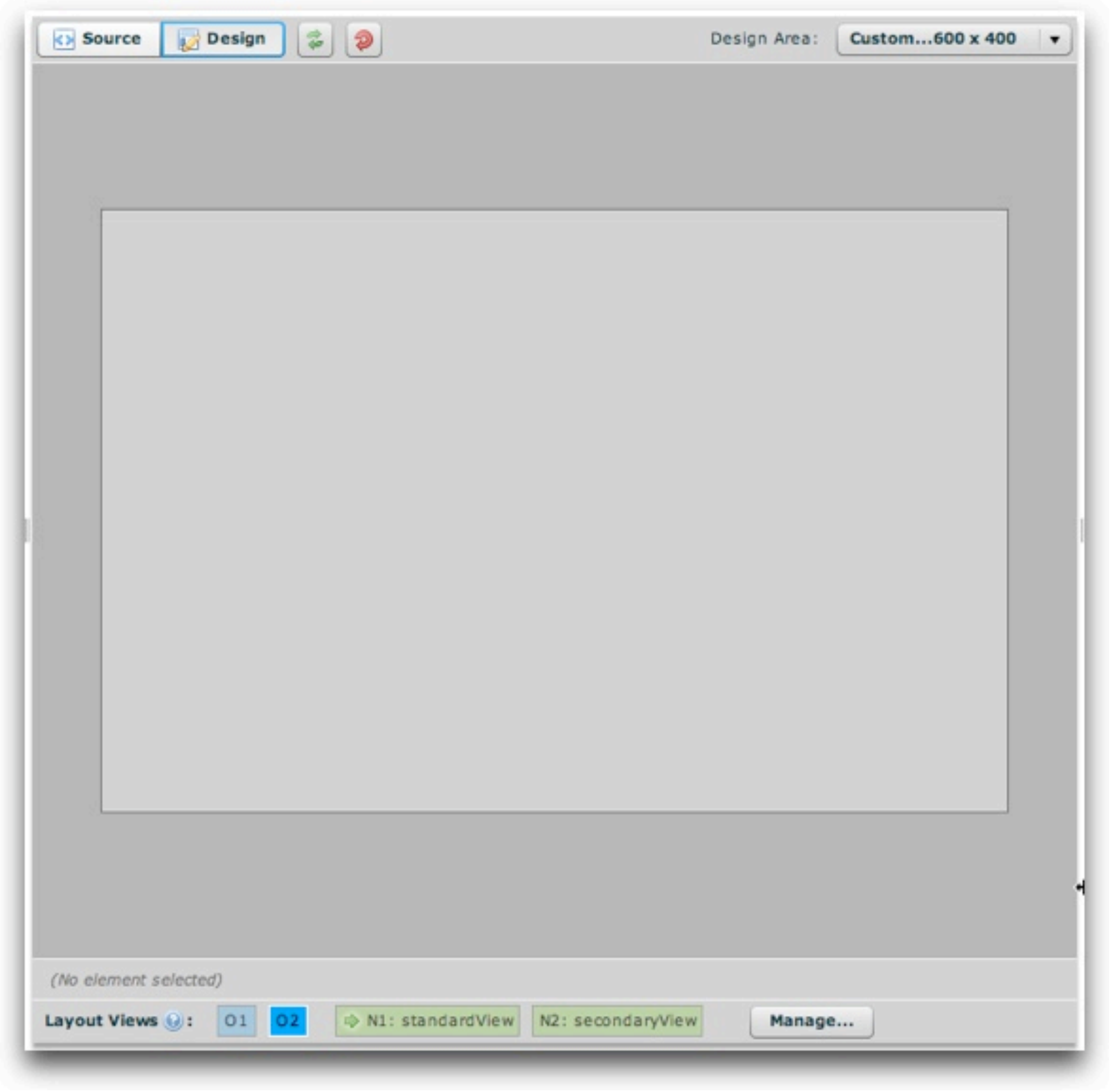
When you're done adding and managing custom elements, click the 'X' in the upper-right corner of the dialog to return to the main wizard. All of your custom elements will be accessible through the 'My Custom Elements' section of the Components Panel tree:



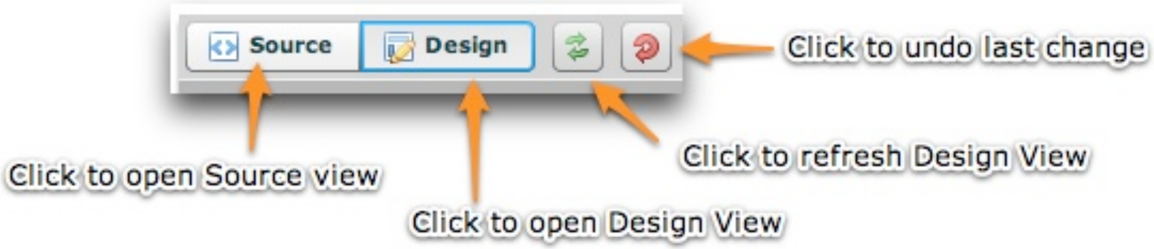
Custom elements can be used like any other built-in element; just drag and drop them into your layouts.

Using the Design Area

The design area is the main area in the center.   It appears like this:

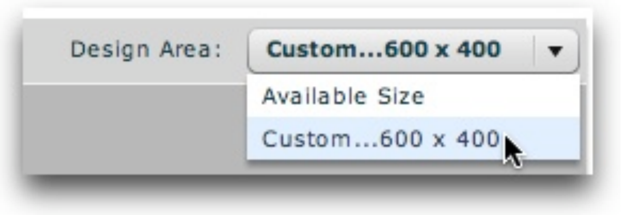


At the top-left are a number of control buttons:

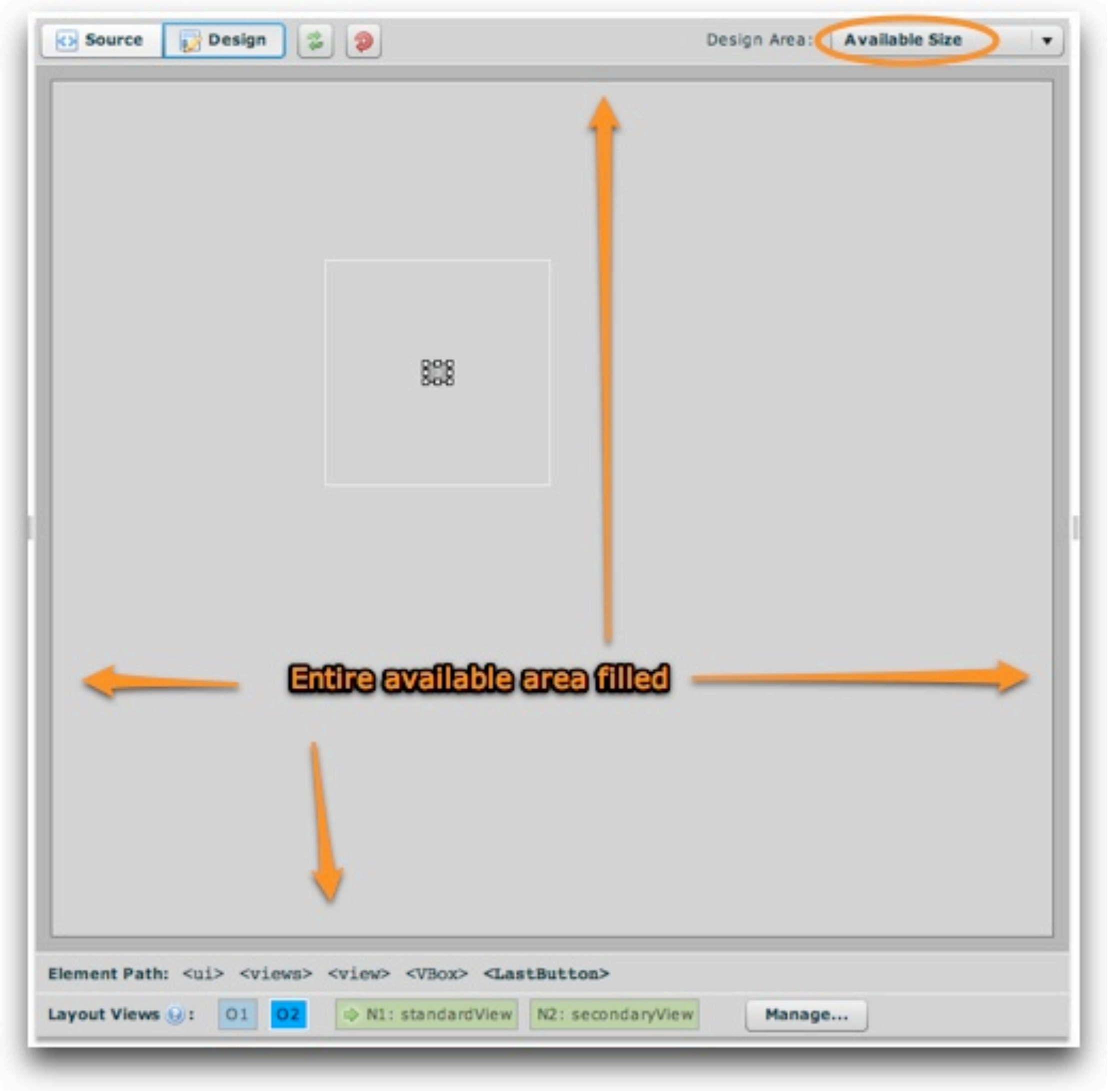

Source
Design
Click to undo last change
Click to refresh Design View
Click to open Source view
Click to open Design View

The 'Source' button opens Source View, where you can edit all of the Layout XML directly. Note -- the other XML code from the configuration file is not shown in this view; only the Layout XML is shown (from the <ui> section of the configuration file).   The 'Design' view button activates the standard WYSIWYG design view.   The Refresh button allows you to refresh the layout rendering if something doesn't seem to be updating properly.   The final button allows you to undo the last change you made to the layout.

On the top-right are the controls for the design area's size:



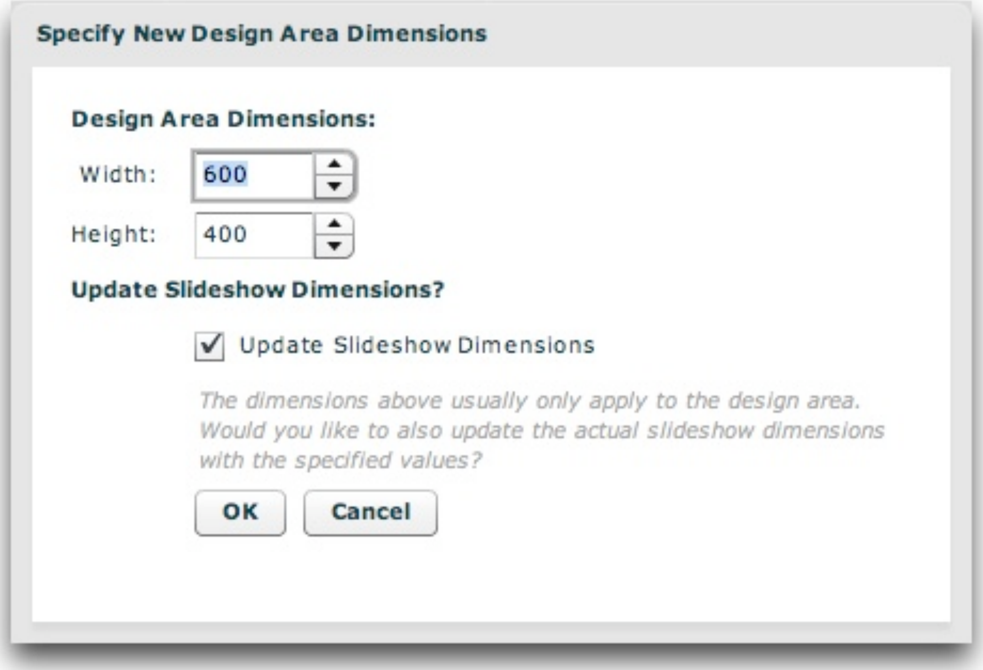
Note that these dimensions only control the appearance of design view in the wizard.   They are separate from the actual slideshow dimensions (specified in the Appearance Settings section).   You can either set the design area to fill the available size:



Entire available area filled

Element Path: <ui> <views> <view> <VBox> <LastButton>

Or, you can use custom dimensions (which by default match the slideshow's dimensions, giving a realistic preview) by selecting the 'Custom...' option. This will open a dialog where you can specify the precise desired dimensions:



Specify New Design Area Dimensions

Design Area Dimensions:

Width: 600

Height: 400

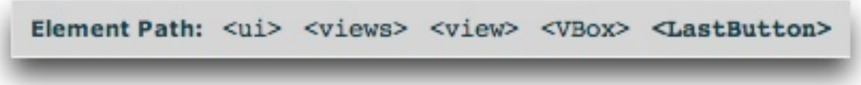
Update Slideshow Dimensions?

☑ Update Slideshow Dimensions

The dimensions above usually only apply to the design area. Would you like to also update the actual slideshow dimensions with the specified values?

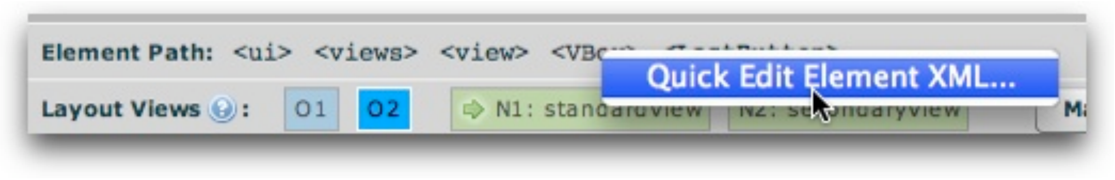
If you want this to be the size of the actual slideshow (in addition to the design view preview), you can specify that the main dimensions should be updated. Note that these custom dimensions are in pixels. As much as possible, you should use a preview size which matches the size of the actual slideshow; this helps to ensure that your layout will appear properly when previewed.

At the bottom of the design area, there are two bars. The first lists the XML path of the currently selected element:

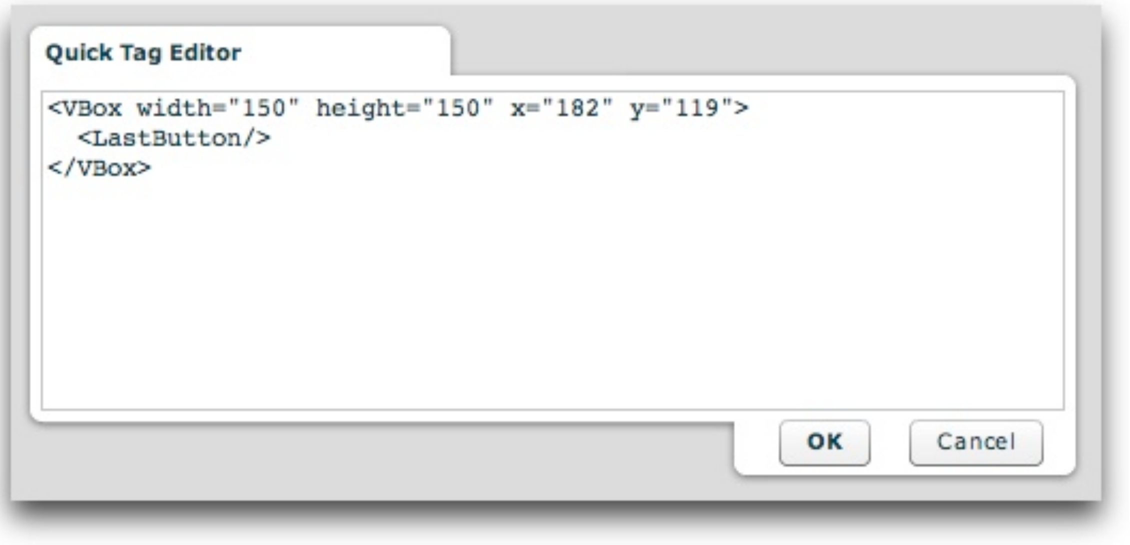


Element Path: <ui> <views> <view> <VBox> <LastButton>

You can select any of the elements in the hierarchy by clicking on its tag name (e.g. click on 'VBox' above to select the LastButton's parent container). You can also edit any element's XML code by right-clicking (Cmd+clicking) on its tag name and selecting 'Quick Edit Element XML':


Element Path: <ui> <views> <view> <VBc
Quick Edit Element XML...
Layout Views ?:
O1
O2
N1: stand
N2: se
Ma

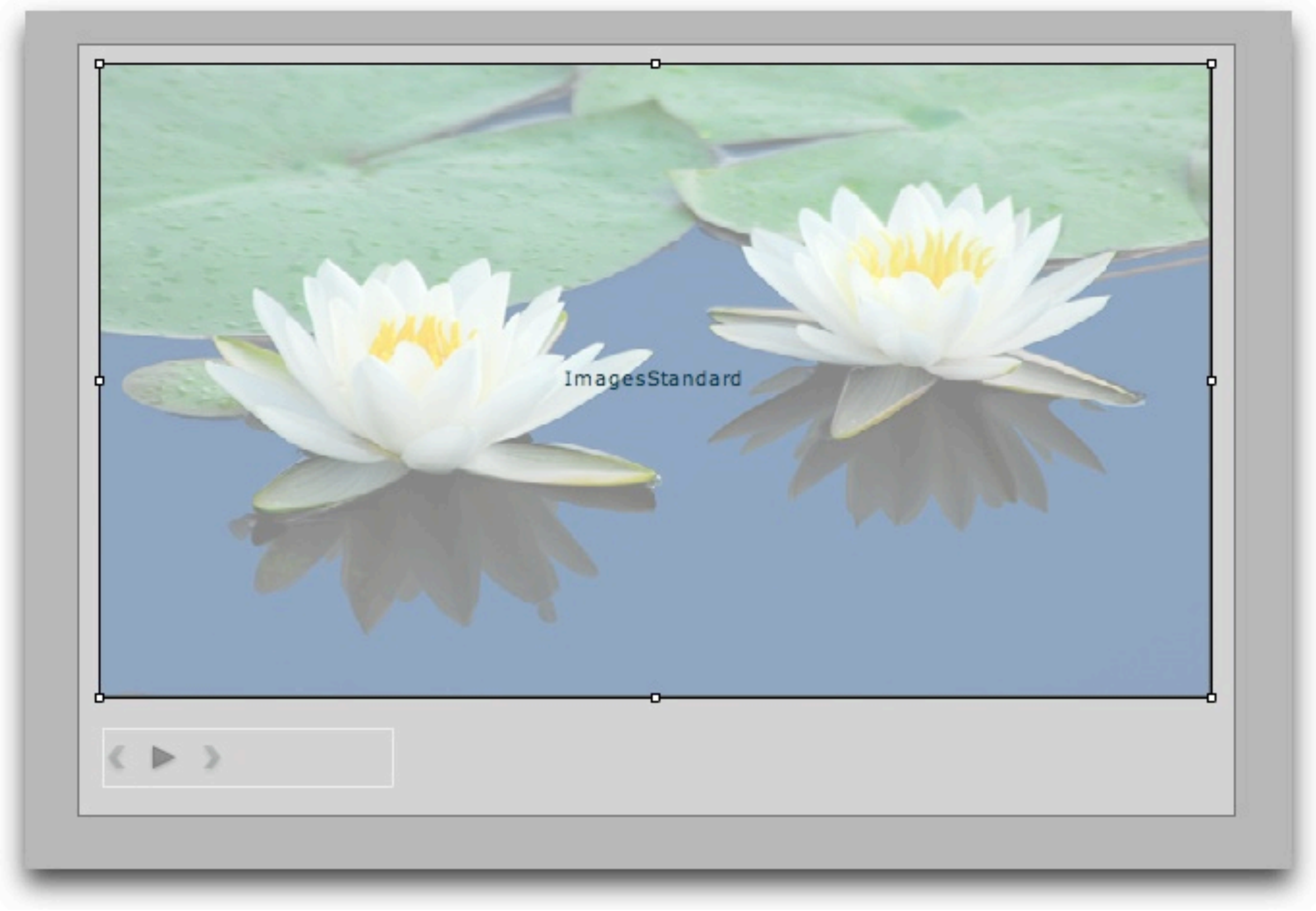
This allows you to edit the XML of that element directly:



```
<VBox width="150" height="150" x="182" y="119">
  <LastButton/>
</VBox>
```

The second bar at the bottom is the Layout Management Bar (be sure to review this topic).   In Design View, only one layout view is shown at a given time. Depending on which view is selected in the Layout Management Bar, only that view will be visible on the stage above.   As mentioned above, you drag and drop elements from the Components Panel onto the stage to add them to that layout view.

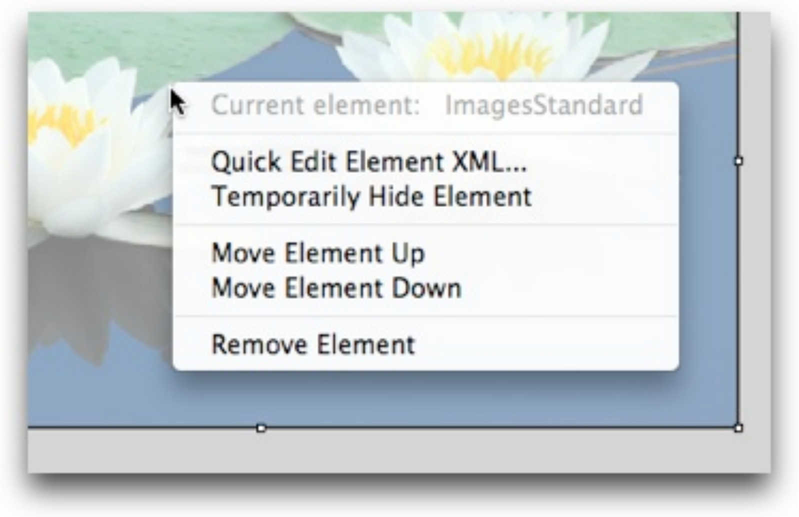
Once you've added elements using design view, your layout view might appear something like this:



To make changes and customizations to an element, you'll use the Properties Panel. However, you must first select the element. There are three methods to select an element:

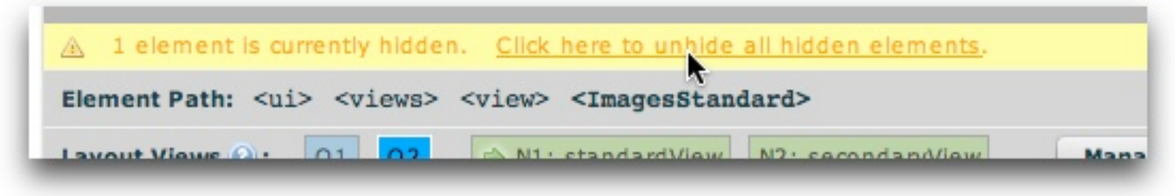
- As mentioned above, you can use the Element Path Bar at the bottom of the design area to select an element.
- You can click on any element in the design area to select it.
- You can use the Outline Panel to select an element. See the topic on Using the Outline Panel for more information.

In the design area, you can perform several element-related tasks by right-clicking (Cmd+clicking) on an element:



The menu will clarify which element is selected (so there is no confusion) and provide the following choices:

- **Quick Edit Element XML** - Launches the Quick Tag Editor as shown above.
- **Temporarily Hide Element** - Allows you to temporarily hide one or more elements in design view. This is helpful if there are multiple elements on top of each other; hiding the upper elements allows access to the lower ones. When one or more elements is hidden, you'll see the following notification bar at the bottom of the design area, allowing you to easily un-hide the hidden elements:



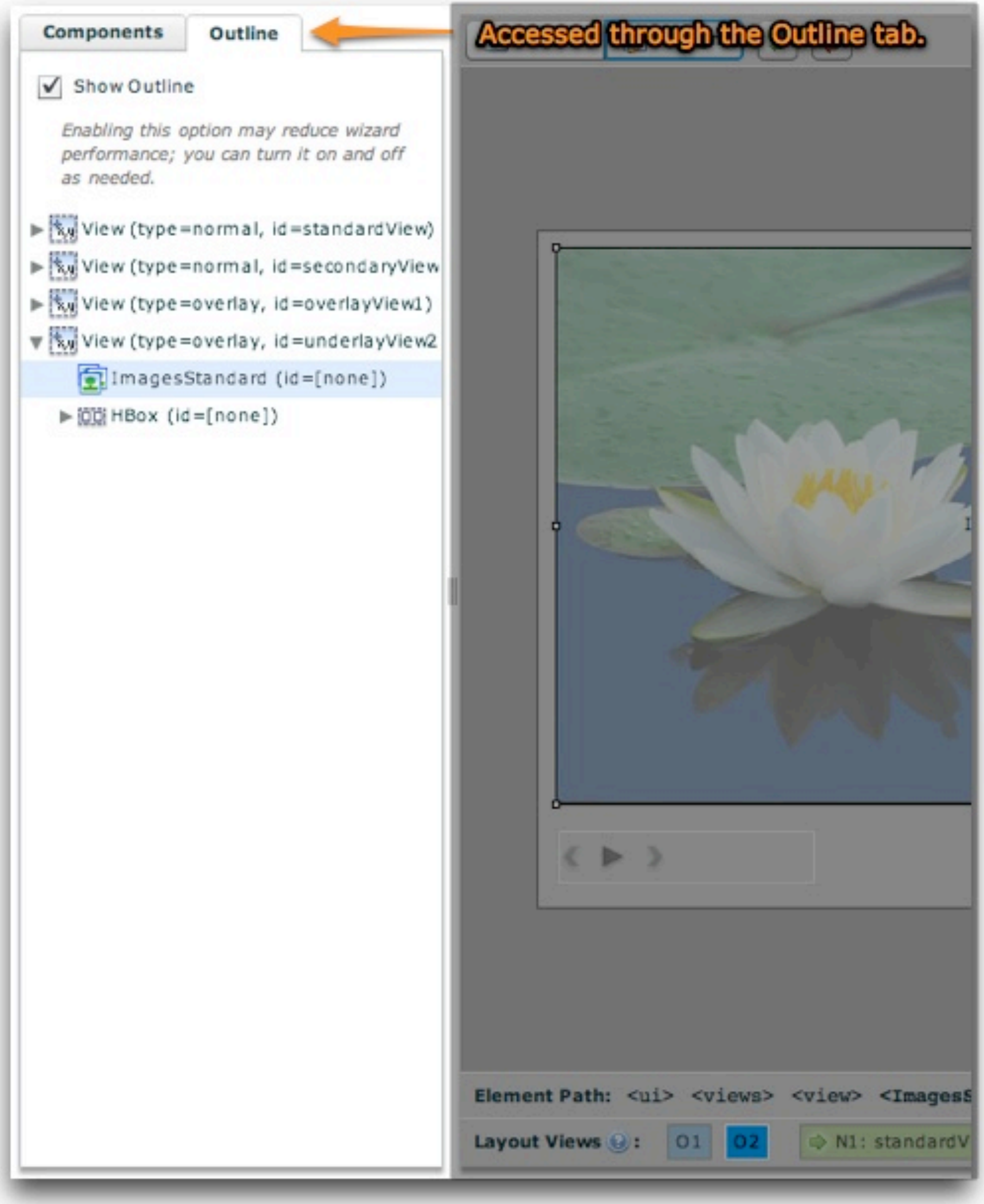
⚠ 1 element is currently hidden. Click here to unhide all hidden elements.

Element Path: <ui> <views> <view> <ImagesStandard>

- **Move Element Up** - Moves the element higher in the visual stacking order, making it appear on top of its sibling elements (elements which are in the same parent container). If you attempt to select this option and the element is already at the top of the stacking order, a notification will tell you that the element cannot be moved higher.
- **Move Element Down** - Moves the element lower in the visual stacking order, making it appear below its sibling elements (elements which are in the same parent container). If you attempt to select this option and the element is already at the bottom of the stacking order, a notification will tell you that the element cannot be moved lower.
- **Remove Element** - Removes the current element from your layout.

Using the Outline Panel

The Outline Panel allows you to see a hierarchical view of all elements in a slideshow's layout. It can be used for easy selection of elements as well as getting a quick overview. The panel is located on the left side of the design area, in the same tab navigator as the Components Panel:



Accessed through the Outline tab.

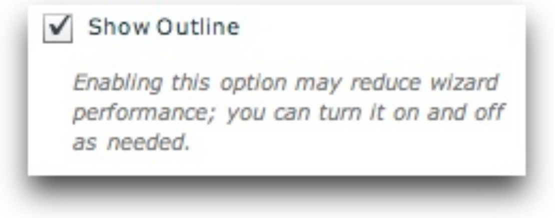
☑ Show Outline

Enabling this option may reduce wizard performance; you can turn it on and off as needed.

```
▶ View (type=normal, id=standardView)
▶ View (type=normal, id=secondaryView
▶ View (type=overlay, id=overlayView1)
▼ View (type=overlay, id=underlayView2
      ImagesStandard (id=[none])
    ▶ HBox (id=[none])
```

Element Path: `<ui> <views> <view> <ImagesS`

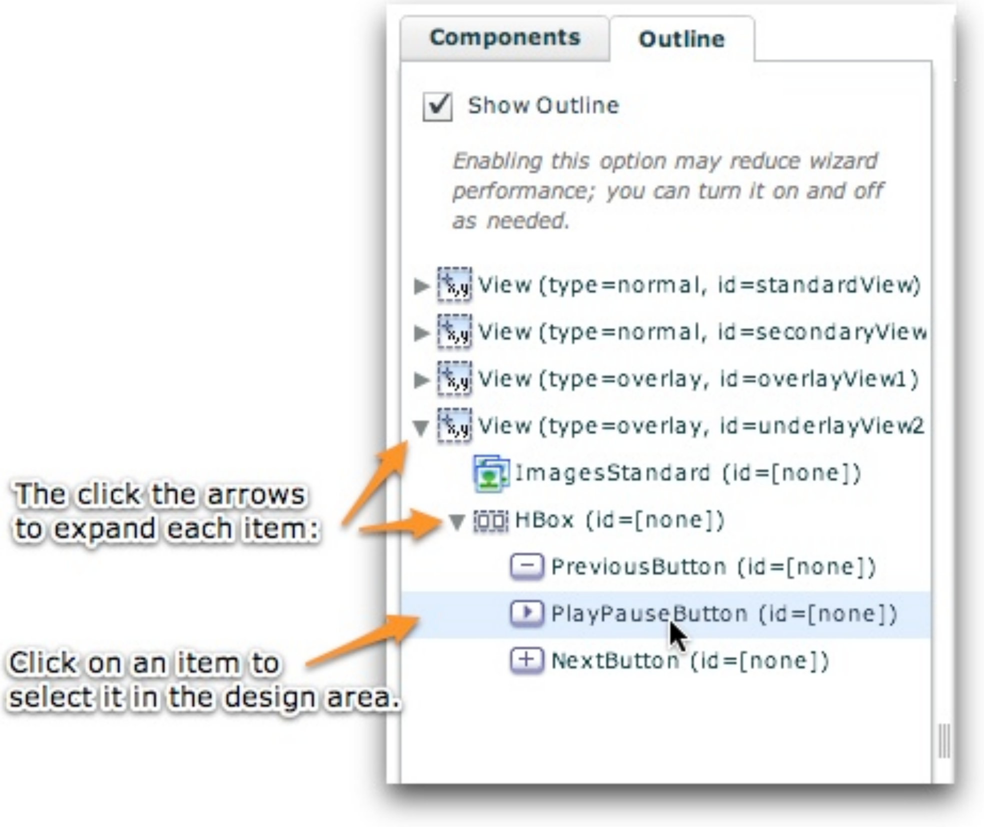
The outline tree should only be enabled when in use, as it may reduce wizard performance. After opening the Outline Panel, you can disable the outline by unchecking the 'Show Outline' checkbox:



☑ Show Outline

Enabling this option may reduce wizard performance; you can turn it on and off as needed.

When enabled, a hierarchical tree view of layout views and their constituent elements is shown. To expand each expandable item, click the gray arrow at its left edge. Clicking on an item will select that element in the design area and allow you to edit its properties in the Properties Panel.



☑ Show Outline

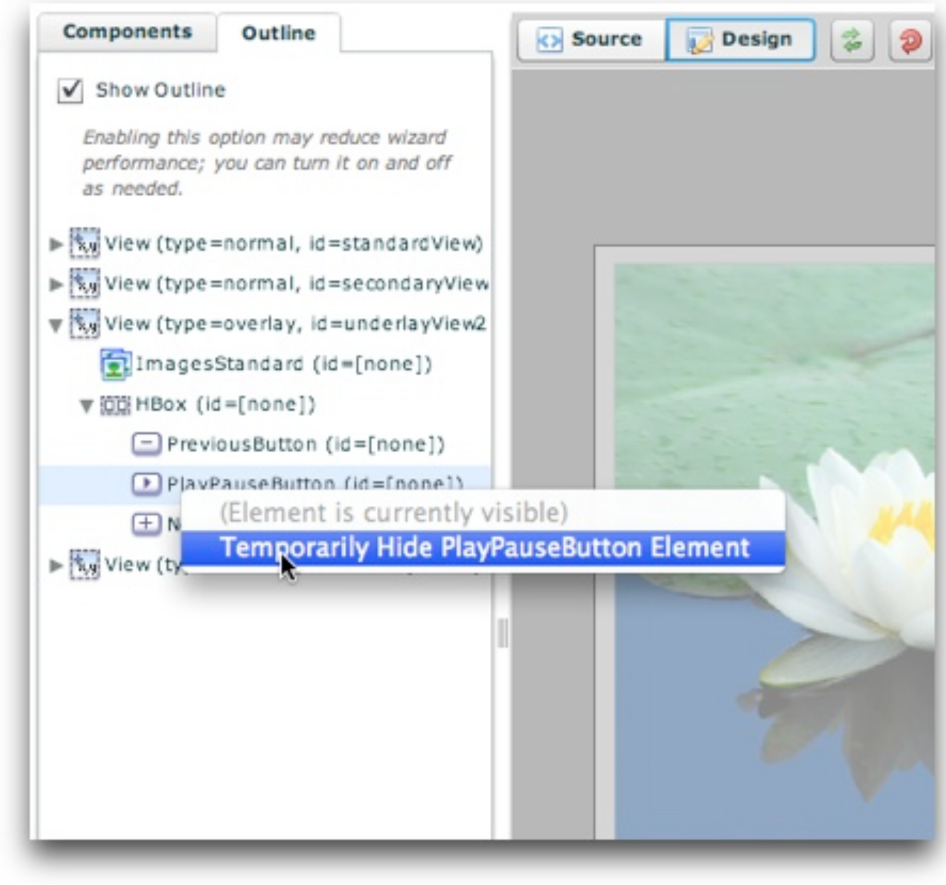
Enabling this option may reduce wizard performance; you can turn it on and off as needed.

▶ View (type=normal, id=standardView)
▶ View (type=normal, id=secondaryView
▶ View (type=overlay, id=overlayView1)
▼ View (type=overlay, id=underlayView2
ImagesStandard (id=[none])
▼ HBox (id=[none])
PreviousButton (id=[none])
PlayPauseButton (id=[none])
NextButton (id=[none])

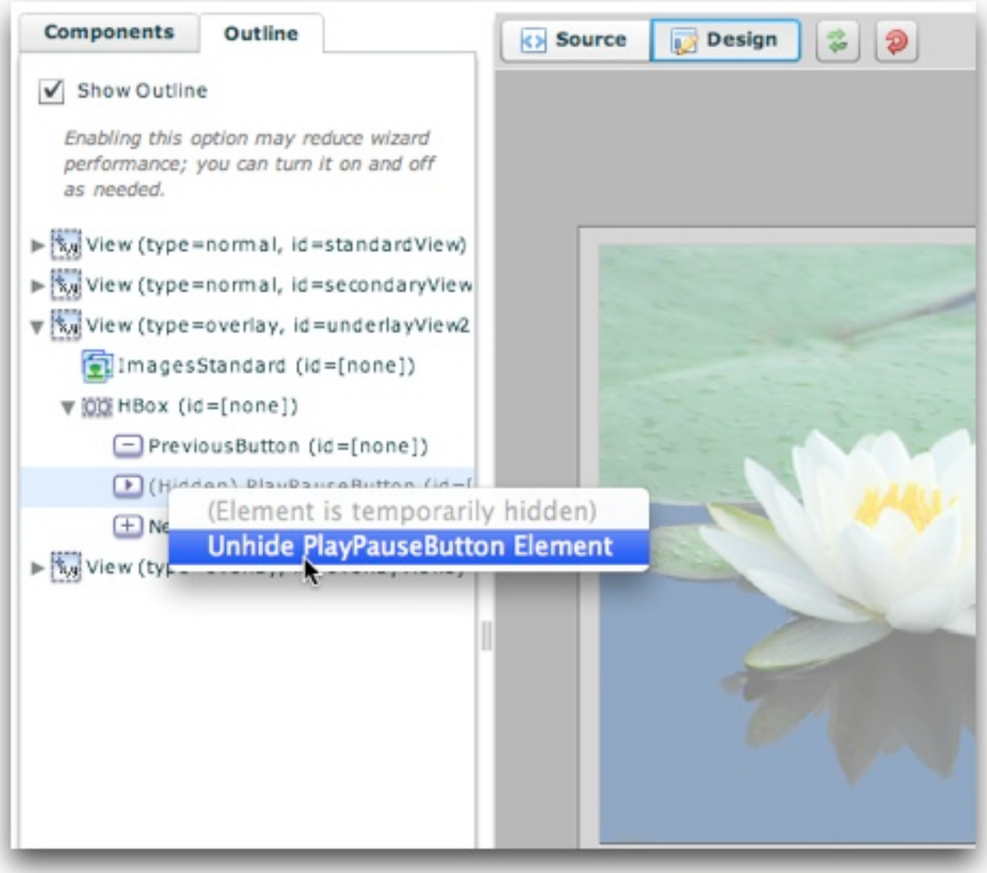
The click the arrows to expand each item:

Click on an item to select it in the design area.

You can also temporarily hide (and unhide) elements from being visible in design view through the outline tree. To do this, right-click (Cmd+click) on an item in the tree and select 'Temporarily Hide *Element Name* Element':


Components
Outline
Source
Design
☑ Show Outline
Enabling this option may reduce wizard performance; you can turn it on and off as needed.
View (type=normal, id=standardView)
View (type=normal, id=secondaryView
View (type=overlay, id=underlayView2
ImagesStandard (id=[none])
HBox (id=[none])
PreviousButton (id=[none])
PlayPauseButton (id=[none])
N
View (ty
(Element is currently visible)
Temporarily Hide PlayPauseButton Element

To unhide a hidden element, select 'Unhide *Element Name* Element':

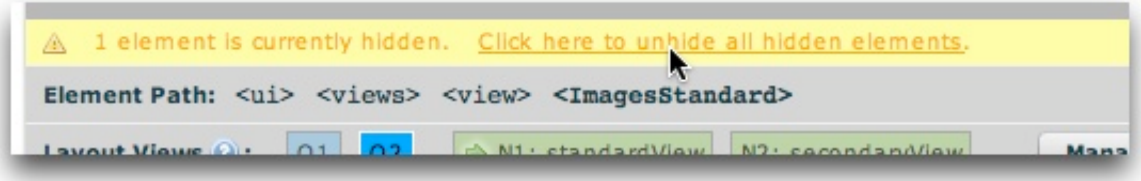


Components
Outline
Source
Design
View (type=normal, id=standardView)
View (type=normal, id=secondaryView
View (type=overlay, id=underlayView2
ImagesStandard (id=[none])
HBox (id=[none])
PreviousButton (id=[none])
(Hidden) PlayPauseButton (id=[
Ne
View (ty
(Element is temporarily hidden)
Unhide PlayPauseButton Element

☑ Show Outline

Enabling this option may reduce wizard performance; you can turn it on and off as needed.

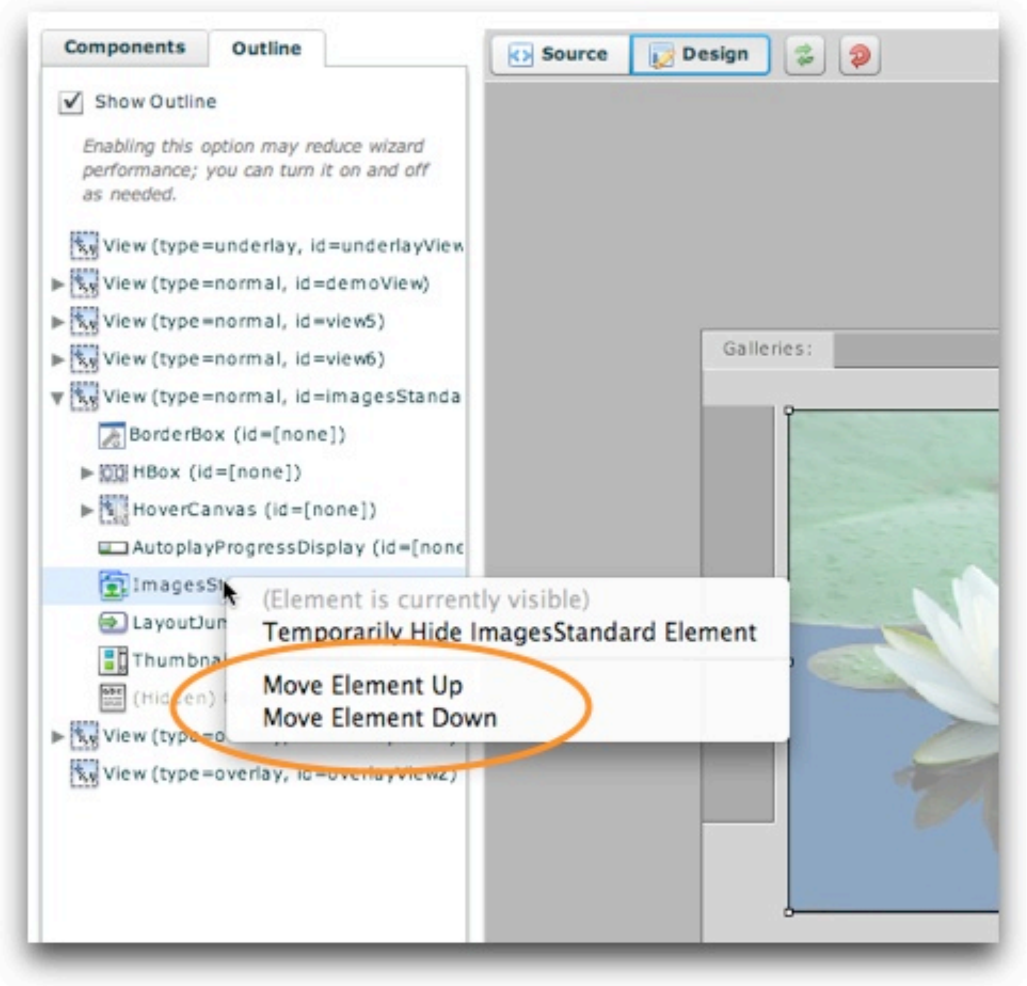
Temporary element hiding is helpful if there are multiple elements on top of each other; hiding the upper elements allows access to the lower ones. When one or more elements is hidden, you'll see the following notification bar at the bottom of the design area, allowing you to easily un-hide the hidden elements:



1 element is currently hidden. Click here to unhide all hidden elements.

Element Path: <ui> <views> <view> <ImagesStandard>

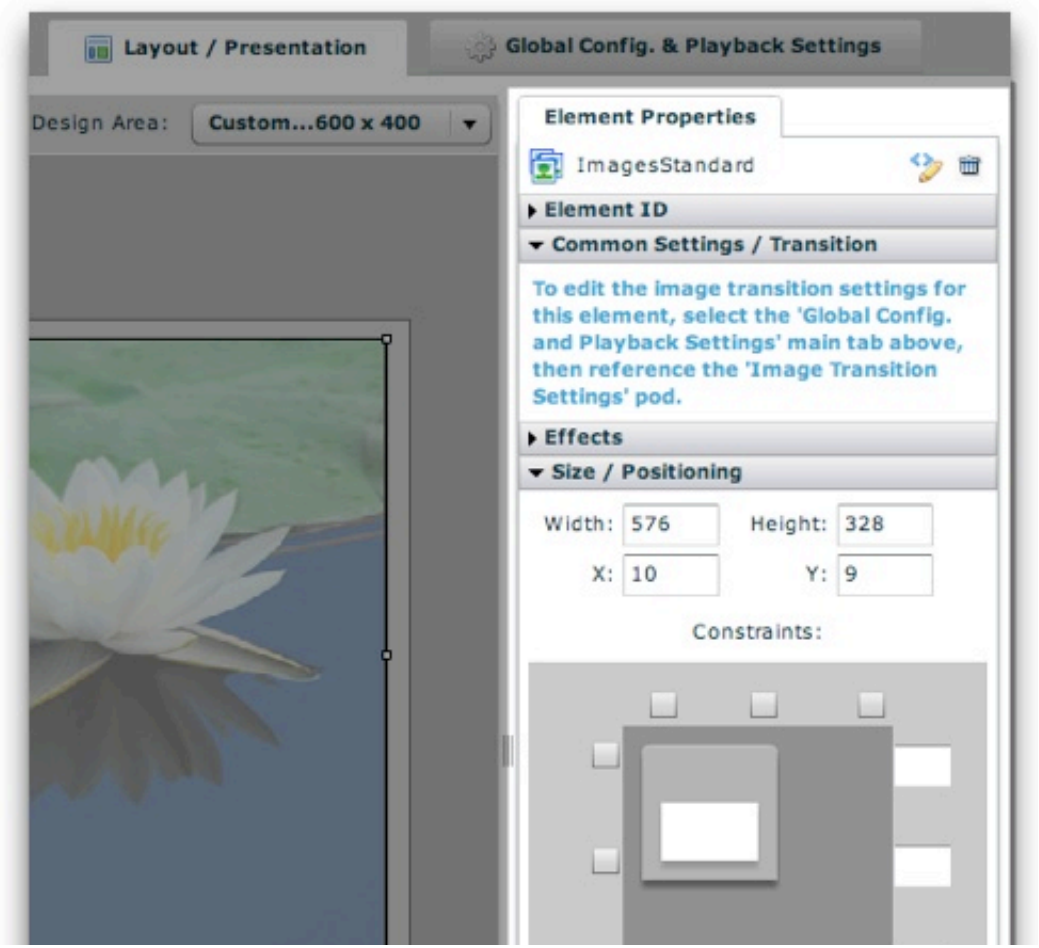
You can also change the visual stacking order of elements by right-clicking (Cmd+clicking) on an element in the tree and selecting Move Item Up or Move Item Down:


Components
Outline
☑ Show Outline
Enabling this option may reduce wizard performance; you can turn it on and off as needed.
View (type=underlay, id=underlayView
View (type=normal, id=demoView)
View (type=normal, id=view5)
View (type=normal, id=view6)
View (type=normal, id=imagesStanda
BorderBox (id=[none])
HBox (id=[none])
HoverCanvas (id=[none])
AutoplayProgressDisplay (id=[none
ImagesSt
LayoutJum
Thumbna
(Hidden)
View (type=o
View (type=overlay, id=overlayView2)
Source
Design
Galleries:
(Element is currently visible)
Temporarily Hide ImagesStandard Element
Move Element Up
Move Element Down

Move Item Up moves an item above its nearest sibling, toward the top of the visual stack (closer to the screen). Conversely, *Move Item Down* moves an item below its nearest sibling, towards the bottom of the visual stack (away from the screen). Note that the bottom of the outline tree represents the top of the visual stack, so *Move Item Up* actually makes the element move one place lower in the tree.

Using the Properties Panel

The Properties Panel contains options for modification of all common settings on each element. It allows you to position, size, configure, and customize each element to fit with your design and behavior goals. The panel is accessed on the right side of the Layout / Presentation section, to the right of the Design Area:



Design Area: Custom...600 x 400

Element Properties

ImagesStandard

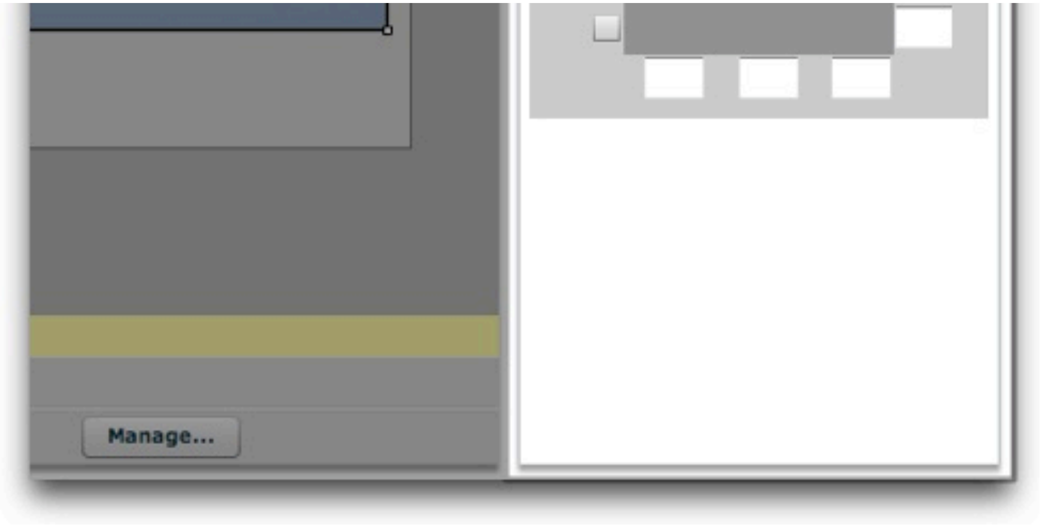
Common Settings / Transition

To edit the image transition settings for this element, select the 'Global Config. and Playback Settings' main tab above, then reference the 'Image Transition Settings' pod.

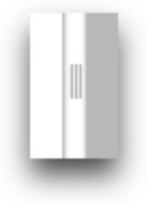
Size / Positioning

Width: 576 Height: 328

X: 10 Y: 9

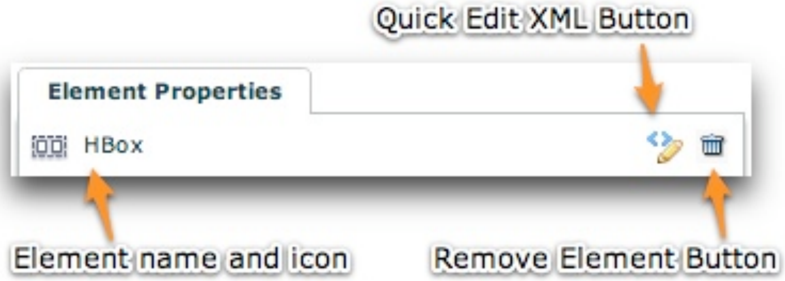


The panel varies for every element based on the available settings. The options for some elements require more width; in these cases, you'll need to expand the width of the panel. To resize the panel, use the draggable resizer on the right edge of the box:

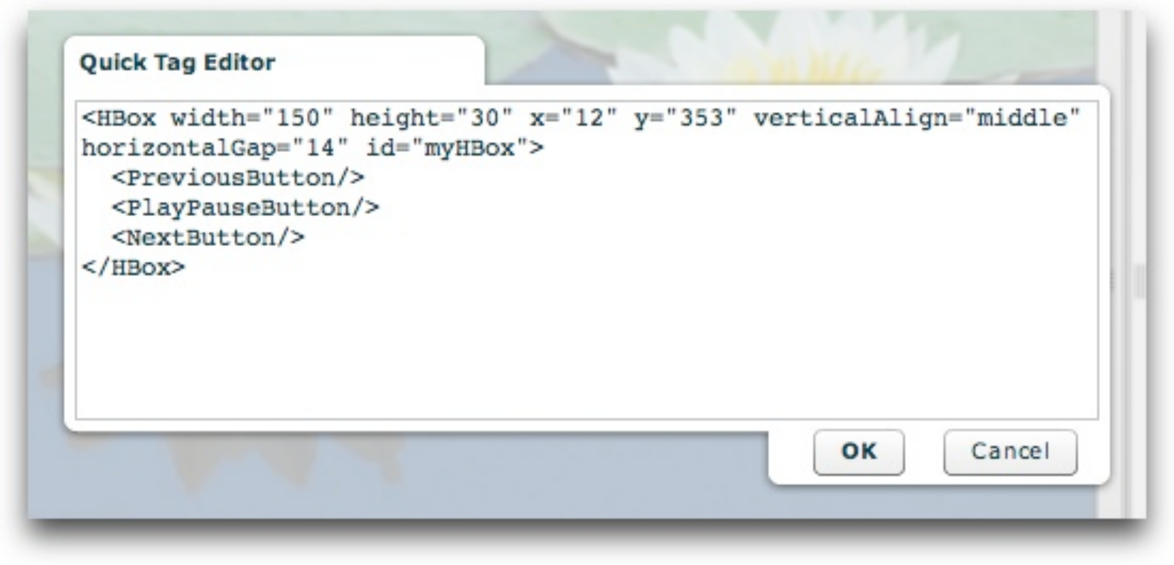


You can shrink it back when finished if you want more room for the design area.

When an element is selected, some common items always appear at the top of the panel:



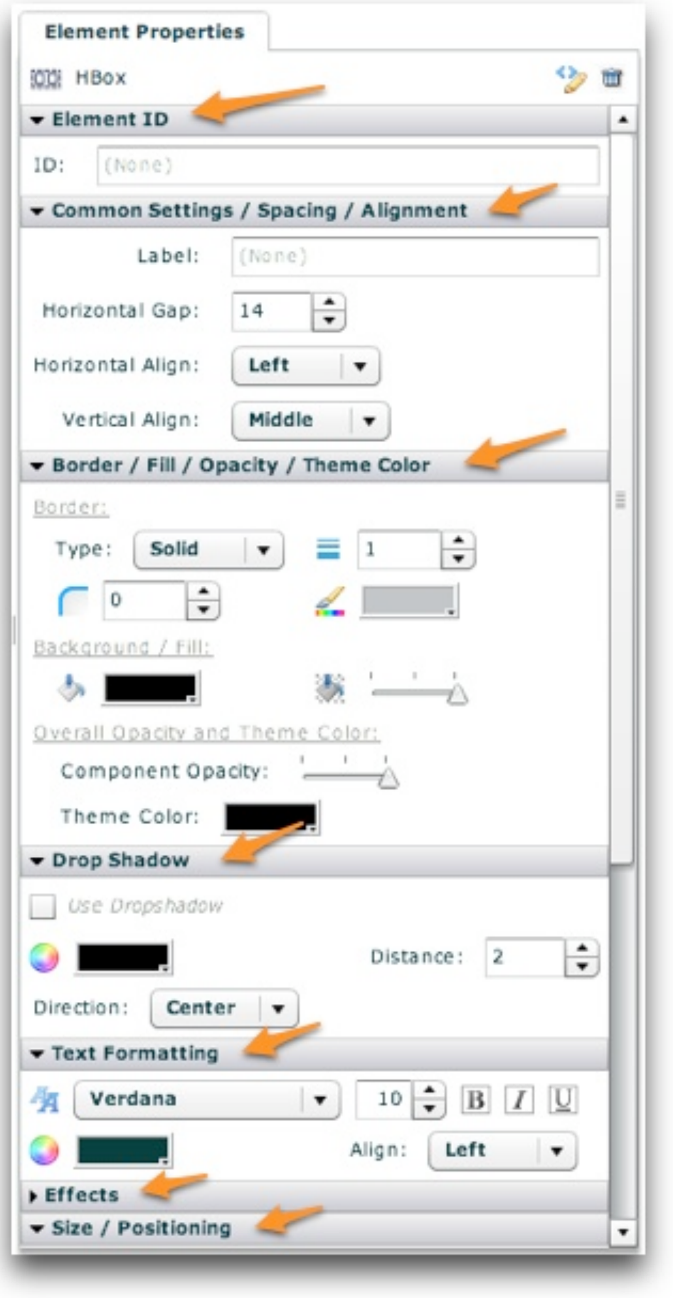
On the left, the element name and icon are displayed. Clicking on the element name will open the element's documentation (in the Available Elements section). The second button on the right is the Remove Element button; clicking it will remove the element from your layout. You can undo a remove by using the Undo button at the top of the Design Area. The first button on the right is the Quick Edit Element XML button. Clicking it will launch the Quick Tag Editor dialog:



```
<HBox width="150" height="30" x="12" y="353" verticalAlign="middle"
horizontalGap="14" id="myHBox">
  <PreviousButton/>
  <PlayPauseButton/>
  <NextButton/>
</HBox>
```

Here, you can make changes directly to the element's XML, including any of its child nodes. Be careful editing the code!

The properties for each element are grouped into expandable/collasible sections, or "pods," within the panel. By default, only some pods will be expanded. To expand a pod, click its title. For example, here an HBox element is shown:



Element Properties

HBox

▾ Element ID

ID: (None)

▾ Common Settings / Spacing / Alignment

Label: (None)

Horizontal Gap: 14

Horizontal Align: Left

Vertical Align: Middle

▾ Border / Fill / Opacity / Theme Color

Border:

Type: Solid 1

0

Background / Fill:

Overall Opacity and Theme Color:

Component Opacity:

Theme Color:

▾ Drop Shadow

□ Use Dropshadow

Distance: 2

Direction: Center

▾ Text Formatting

Verdana 10 B I U

Align: Left

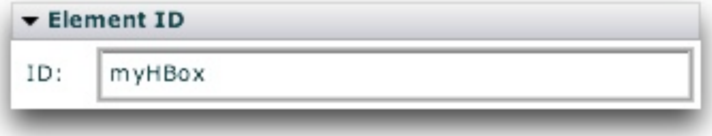
▸ Effects

▾ Size / Positioning

The HBox has so many properties that they don't even all show on this screenshot. Its spacing and alignment, it border and background and dropshadow, and its text styles can be set (the text styles apply to all of its child elements). Every element has a different set of options, but there are some common settings which all (or almost all) elements have. These are listed below:

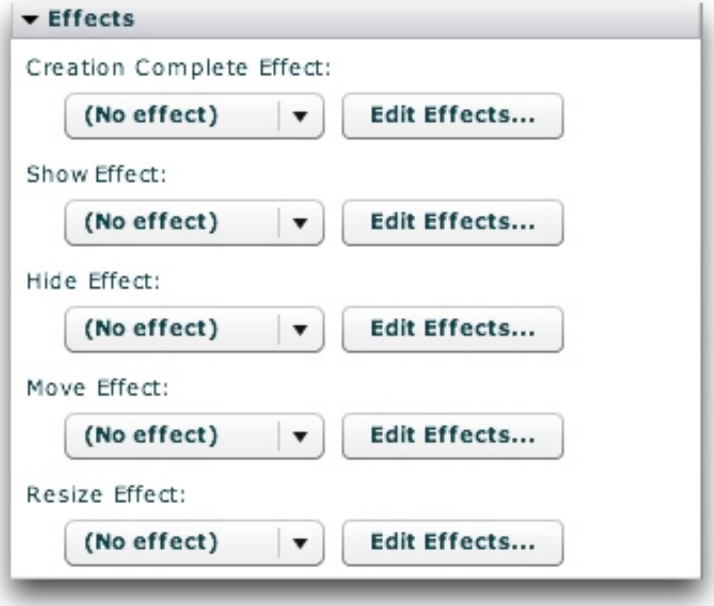
Settings Common To All Elements

First, there is the **Element ID value**. This optional value specifies a unique ID which can be used to reference the component elsewhere to set styles or properties dynamically at runtime (e.g. via a button click):



ID: myHBox

Next, there is the **Effects section**. Effects can be applied to various events in an element's lifecycle. To learn more about using UI Effects, see the Using UI Effects topic.



Creation Complete Effect: (No effect)

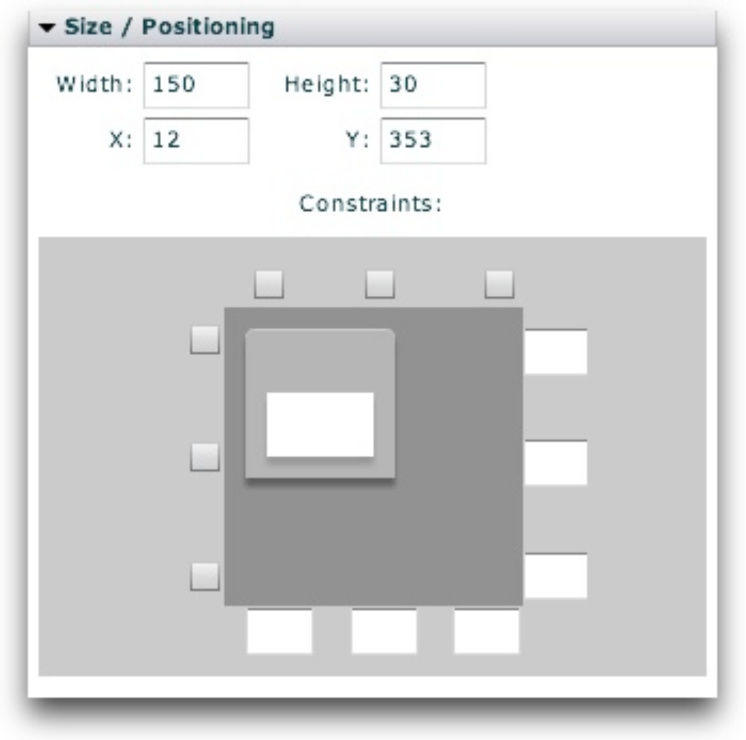
Show Effect: (No effect)

Hide Effect: (No effect)

Move Effect: (No effect)

Resize Effect: (No effect)

Finally, there is the **Size / Position section**. This is where you can precisely position and size the element. To learn more about element positioning and sizing, see the Element Positioning and Sizing topic.

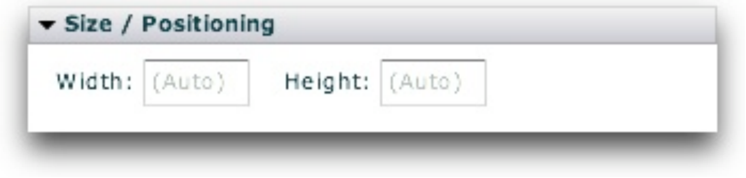


Size / Positioning

Width: 150 Height: 30

X: 12 Y: 353

Or (depending on the element's container type):



Size / Positioning

Width: (Auto) Height: (Auto)

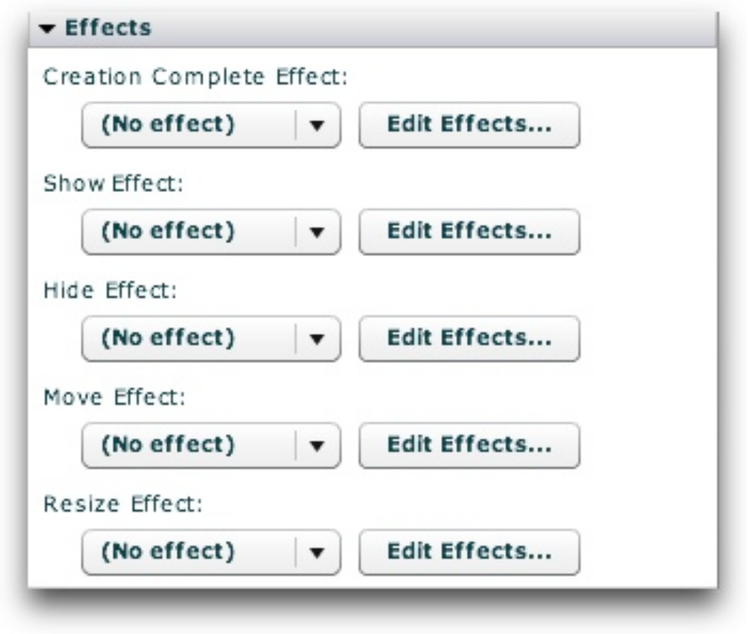
Using the External Wizard to Create and Edit Slideshows > Section-by-Section Usage Guide > Using the 'Layout / Presentation' Section > **Using UI Effects**

Using UI Effects

UI Effects allow you to add special effects to events which occur during a component's lifecycle. The following types of effects are defined for almost all elements (all may not be applicable to a specific element):

- **Creation Complete Effect** - Specifies the effect applied to the element upon its creation.
- **Show Effect** - Specifies the effect applied to the element when it is revealed/showed (made visible by setting `visible` to `true`).
- **Hide Effect** - Specifies the effect applied to the element when it is hidden (`visible` set to `false`).
- **Move Effect** - Specifies the effect applied to the element when it is repositioned / moved.
- **Resize Effect** - Specifies the effect applied to the element when it is resized (when its dimensions change).

Effects are applied using the Effects section of the Properties Panel:



Creation Complete Effect: (No effect)

Show Effect: (No effect)

Hide Effect: (No effect)

Move Effect: (No effect)

Resize Effect: (No effect)

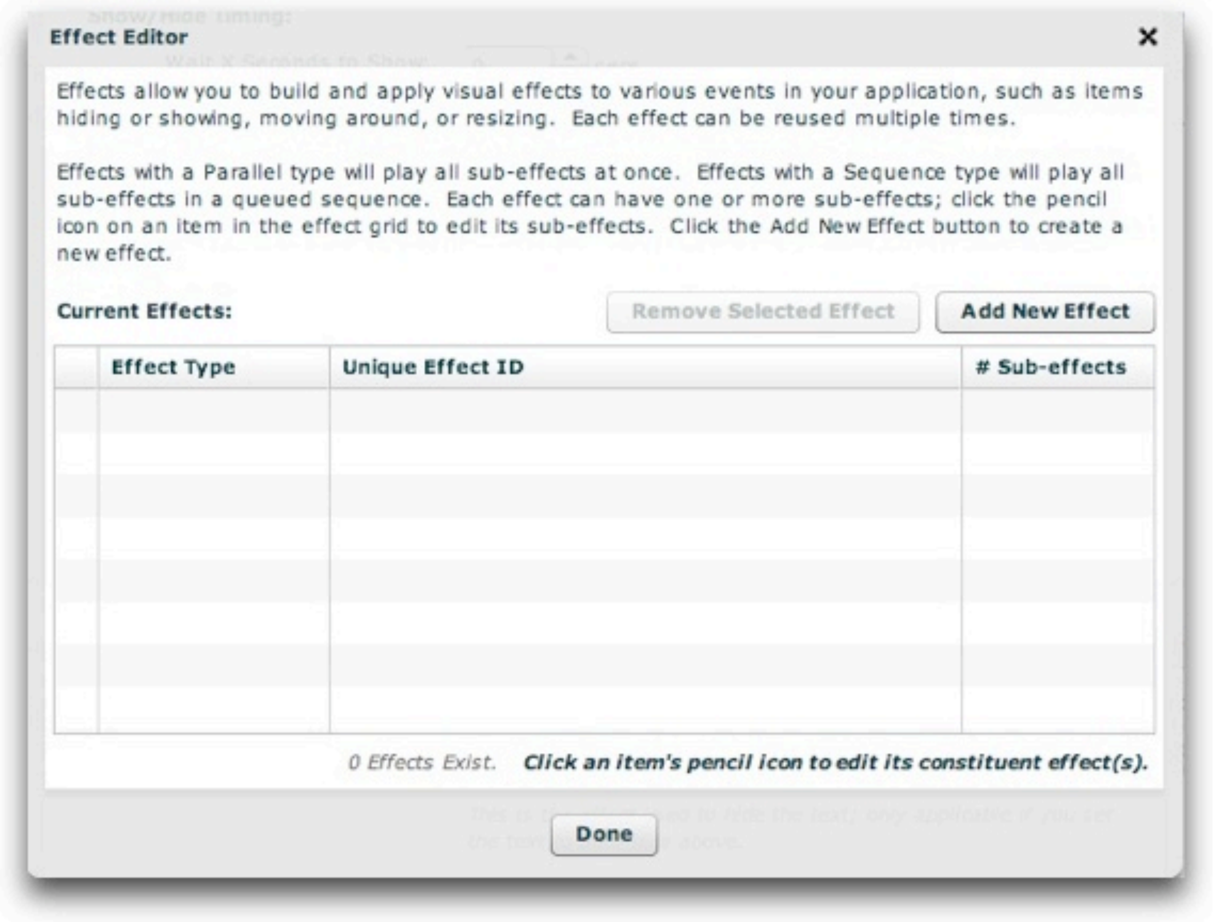
To apply an effect, you must first define an effect to use. This is done through the Effect Editor, accessed through the 'Edit Effects...' button. Once you have defined an effect, select its ID value from the appropriate dropdown menu.

Using the Effect Editor

Usage of the Effect Editor is covered in the Using the Effect Editor subtopic.

Using the Effect Editor

The Effect Editor is where you define all user interface (UI) effects which are used within the slideshow. The editor is accessed through various 'Edit Effects...' buttons, located next to effect selectors throughout the wizard. The dialog appears as follows:



Effect Editor

Effects allow you to build and apply visual effects to various events in your application, such as items hiding or showing, moving around, or resizing. Each effect can be reused multiple times.

Effects with a Parallel type will play all sub-effects at once. Effects with a Sequence type will play all sub-effects in a queued sequence. Each effect can have one or more sub-effects; click the pencil icon on an item in the effect grid to edit its sub-effects. Click the Add New Effect button to create a new effect.

Current Effects:

	Effect Type	Unique Effect ID	# Sub-effects

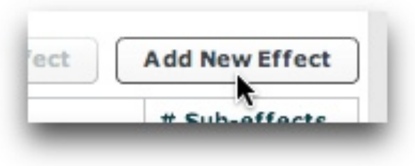
0 Effects Exist. ***Click an item's pencil icon to edit its constituent effect(s).***

Effects allow you to build and apply visual effects to various events in your application, such as items hiding or showing, moving around, or resizing. Each effect is a reusable "package" with a unique ID, and can be referenced by this ID multiple times throughout your slideshow. For example, you could define a simple fade-in effect and a simple fade-out effect. Then, whenever you wanted a fade-in or fade-out, you would reference one of these effects. By centrally defining all effects, unnecessary duplication is avoided.

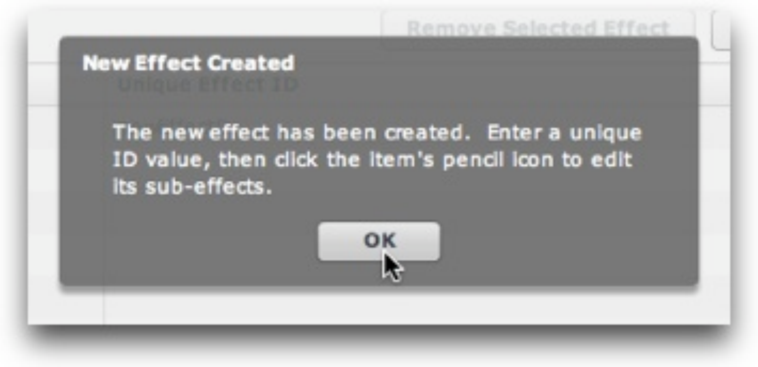
Effects are made up of one or more constituent sub-effects, though most effects contain only one sub-effect. Two different types of effects exist. They only differ if there is more than one sub-effect defined for the effect: Effects with a **Parallel** type will play all sub-effects at once; effects with a **Sequence** type will play all sub-effects in a queued sequence.

Creating a New Effect

To create a new effect, click the Add New Effect button:



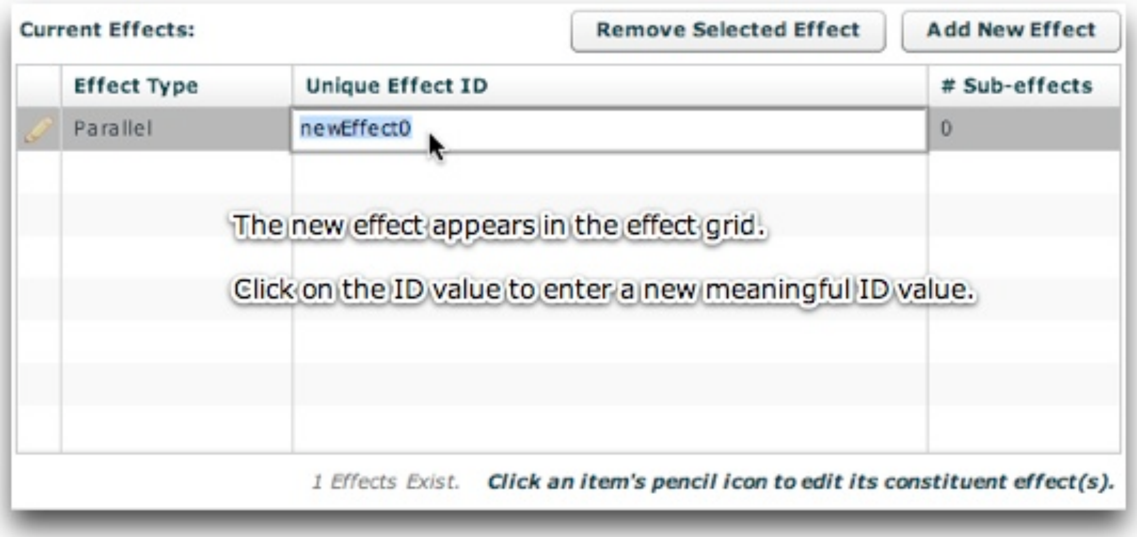
You will be notified that the effect has been created:



New Effect Created

The new effect has been created. Enter a unique ID value, then click the item's pencil icon to edit its sub-effects.

The new effect will appear in the effect grid. First, enter a unique and meaningful ID value. You will use this value to reference the effect elsewhere in the slideshow, so use a value which will help you identify it later:



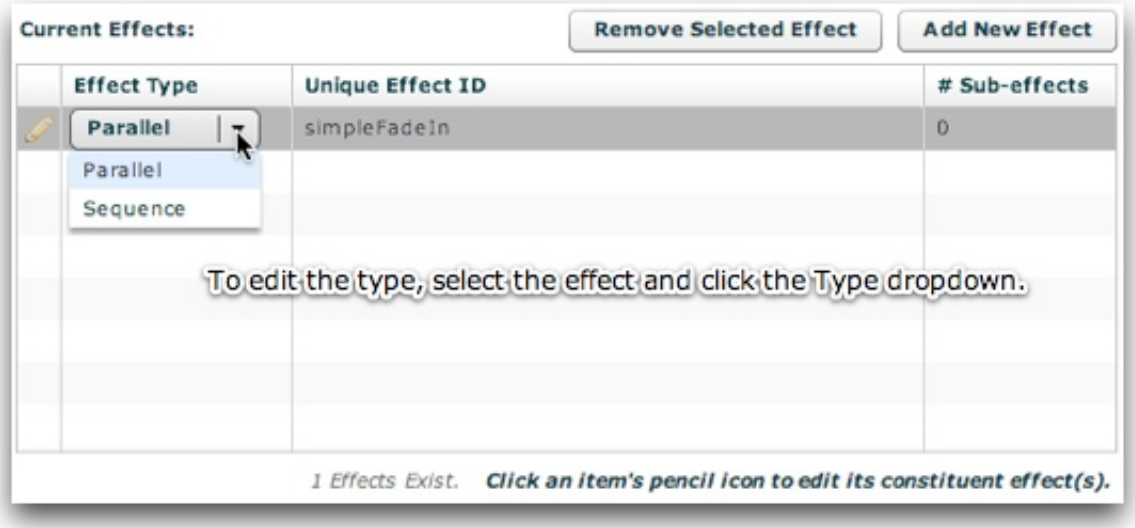
Effect Type	Unique Effect ID	# Sub-effects
Parallel	newEffect0	0

The new effect appears in the effect grid.

Click on the ID value to enter a new meaningful ID value.

1 Effects Exist. *Click an item's pencil icon to edit its constituent effect(s).*

As mentioned above, there are two different types of effects - Parallel and Sequence (see above for the description of each). You can modify the type of a given effect by selecting it and then using the dropdown on the left:



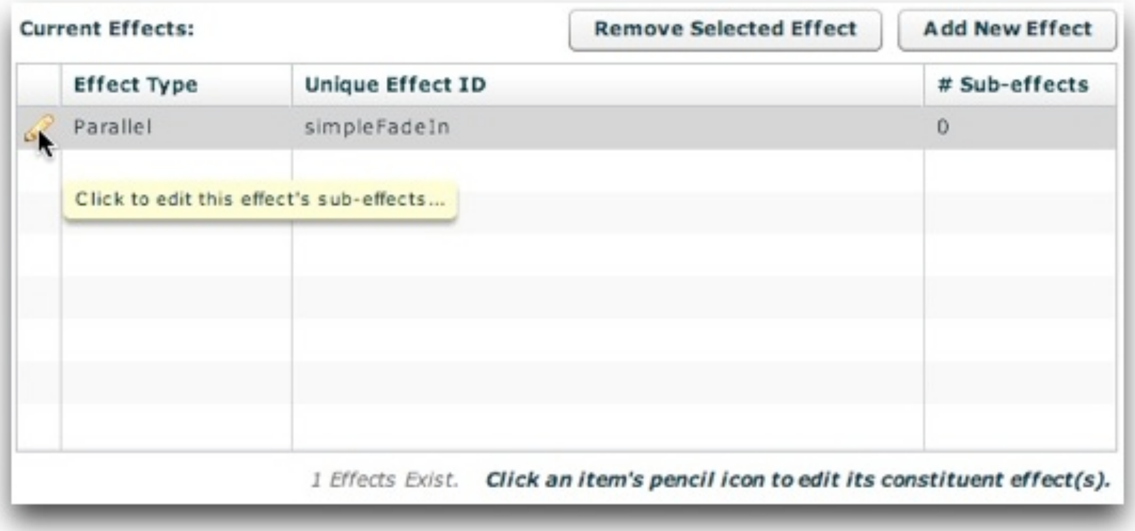
Effect Type	Unique Effect ID	# Sub-effects
Parallel	simpleFadeIn	0

To edit the type, select the effect and click the Type dropdown.

1 Effects Exist. *Click an item's pencil icon to edit its constituent effect(s).*

Managing an Effect's Sub-effects

Once the effect has been created, you must defined at least one constituent sub-effect. This happens in the Edit Sub-effects dialog, which is accessed by clicking an effect's pencil icon in the Effect Editor:



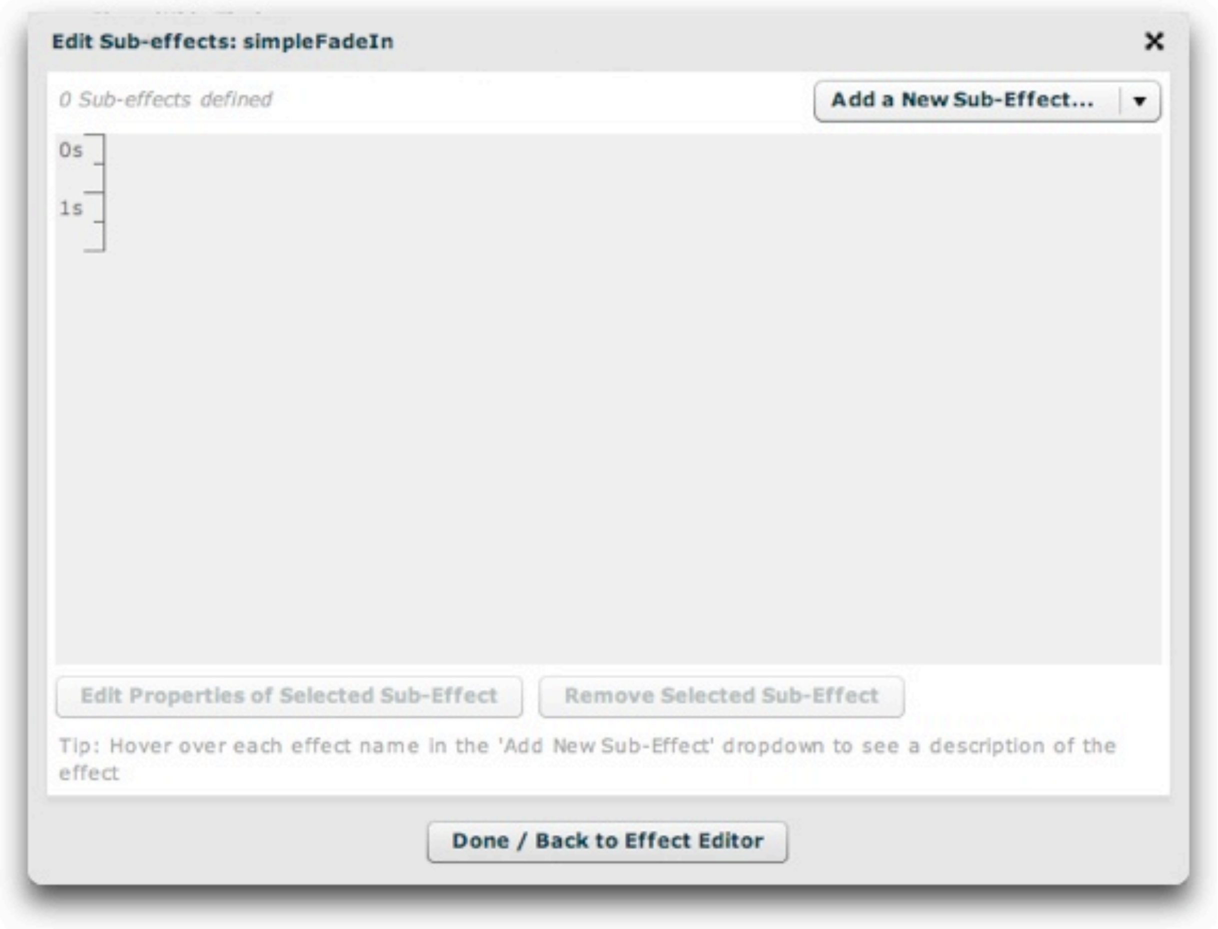
	Effect Type	Unique Effect ID	# Sub-effects
✎	Parallel	simpleFadeIn	0

Click to edit this effect's sub-effects...

1 Effects Exist. ***Click an item's pencil icon to edit its constituent effect(s).***

The appearance of the Edit Sub-effects window varies slightly based on whether the effect's type is Parallel or Sequence. In both instances, the large gray area in the center of the window is where the graphical representations of sub-effects appear.

For a Parallel effect, the time axis is oriented vertically. The graphical representation box for each sub-effect has a fixed width, and its height represents its individual duration:



Edit Sub-effects: simpleFadeIn

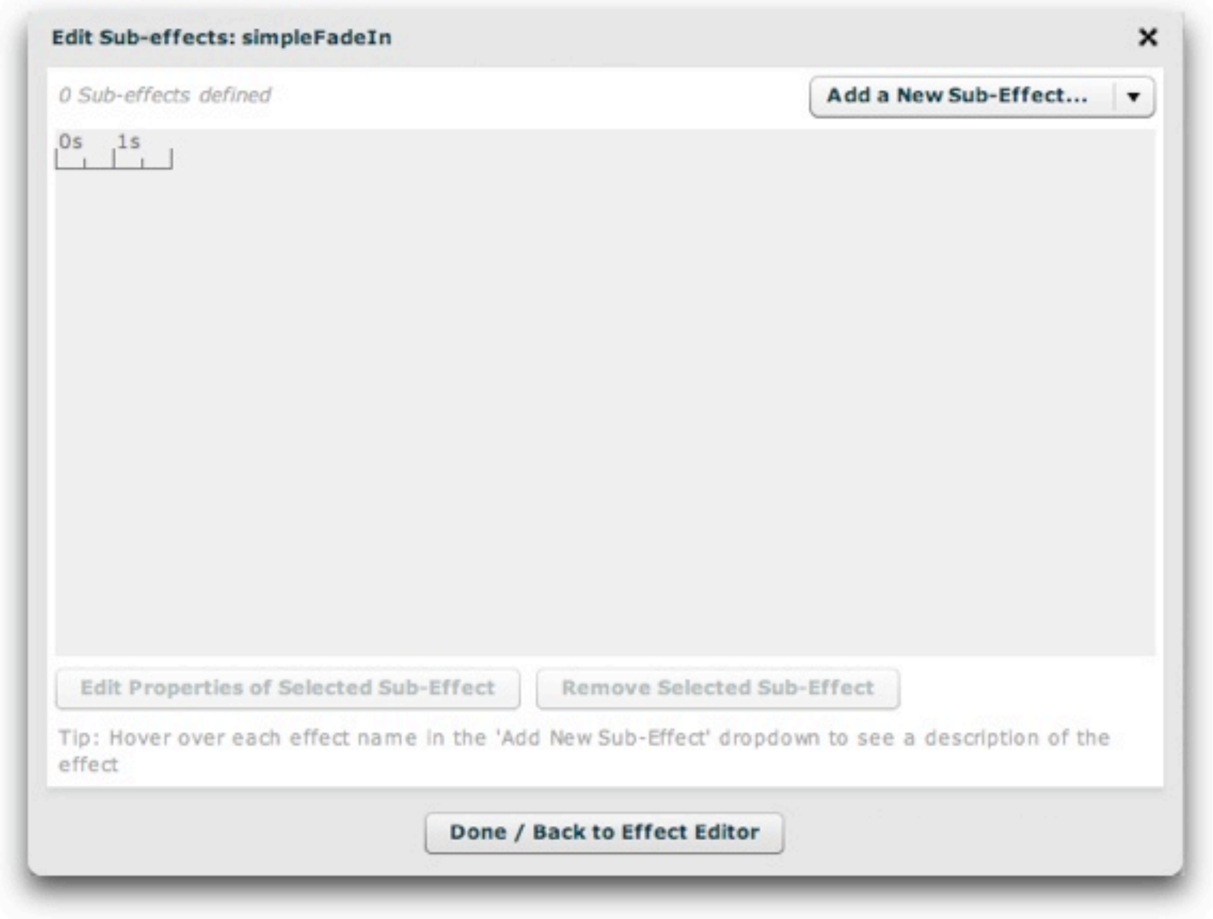
0 Sub-effects defined

×
Add a New Sub-Effect... ▾
0s
1s
Edit Properties of Selected Sub-Effect
Remove Selected Sub-Effect

Tip: Hover over each effect name in the 'Add New Sub-Effect' dropdown to see a description of the effect

Done / Back to Effect Editor

For a Sequence effect, the time axis is oriented horizontally. The graphical representation box for each sub-effect has a fixed height, and its width represents its individual duration:



Edit Sub-effects: simpleFadeIn

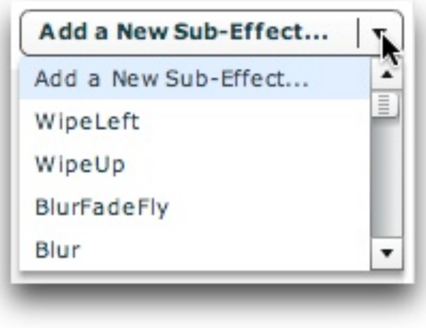
0 Sub-effects defined

Add a New Sub-Effect...
0s
1s
Edit Properties of Selected Sub-Effect
Remove Selected Sub-Effect

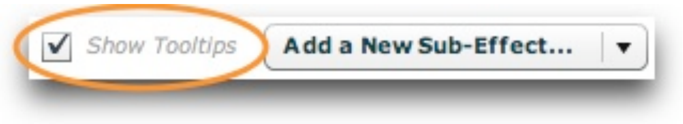
Tip: Hover over each effect name in the 'Add New Sub-Effect' dropdown to see a description of the effect

Done / Back to Effect Editor

To get started, you need to add the first sub-effect. To do so, click the 'Add a New Sub-Effect...' dropdown and select an item:



To see a tooltip description of each effect within the wizard, hover over its name. A detailed summary is shown in the table below. If you prefer to not have the tooltips, you can disable them by unchecking the 'Show Tooltips' checkbox just to the left of the dropdown:



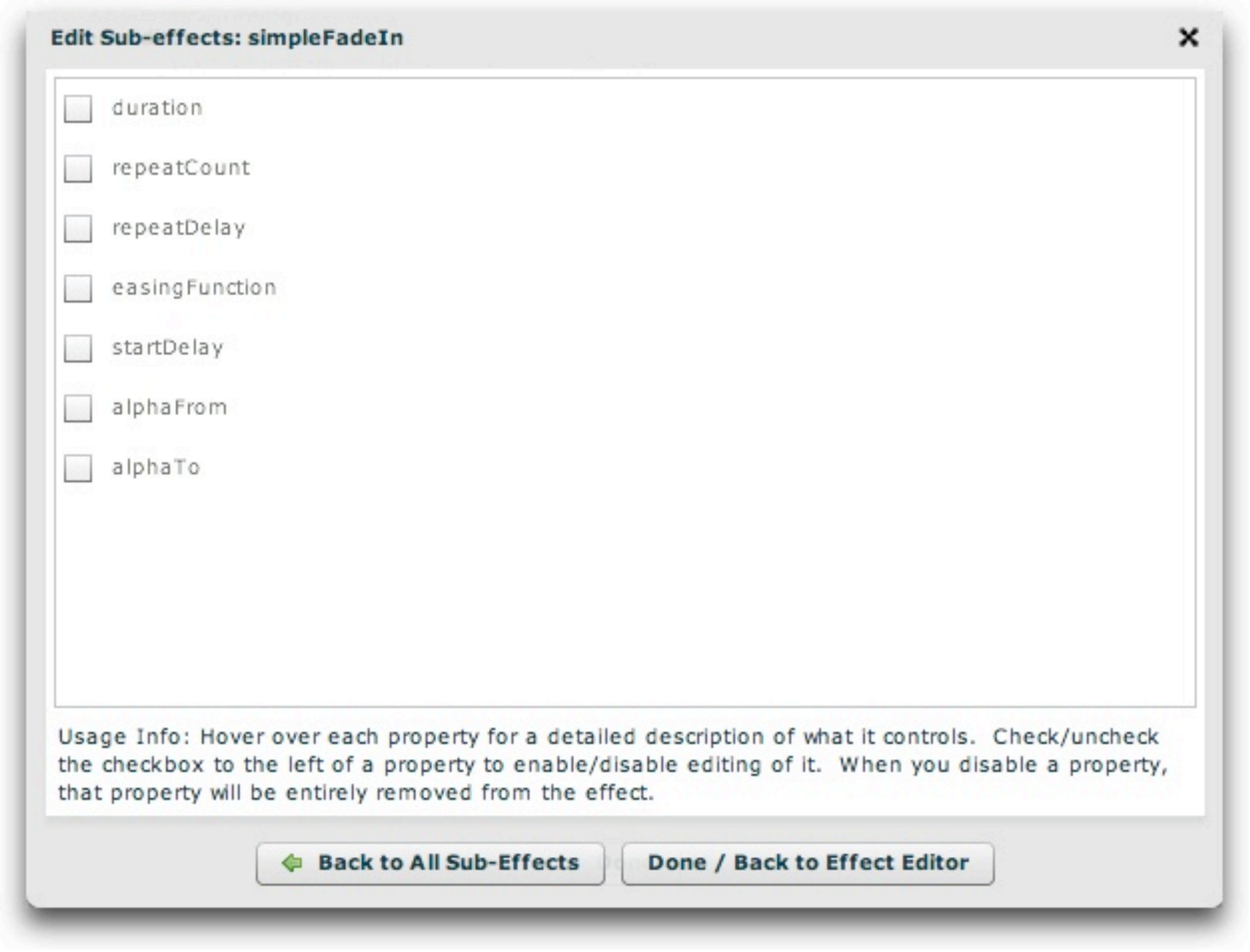
Note: Only some screenshots show the tooltip-toggle checkbox, as this feature was added in a later build.

Expand / Collapse Detailed Sub-effects Summary

Tip for Advanced Users: To learn in more detail about all of the available attributes (properties) for each sub-effect, see the various effect pages which are referenced in the `com.dwuser.ss4.managers.EffectLibrary` class documentation.

Tip: Some of the most commonly-used subeffects include Fade, Blur, and Move.

Once you've added the effect, you will be transitioned to the edit-subeffect-properties view. This view is different for every effect, based on its various parameters. In this example, a Fade effect was added:



□ duration

□ repeatCount

□ repeatDelay

□ easingFunction

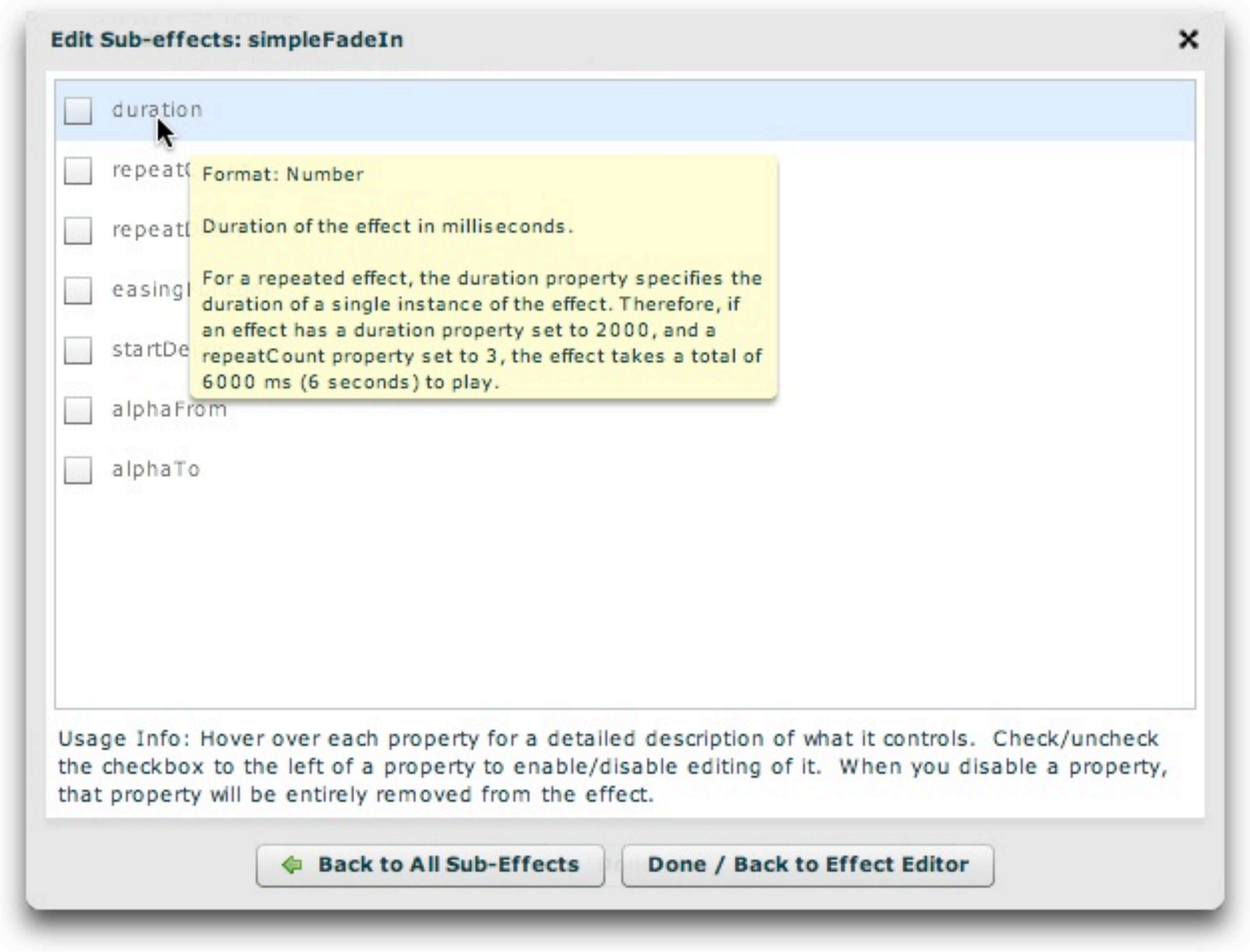
□ startDelay

□ alphaFrom

□ alphaTo

Usage Info: Hover over each property for a detailed description of what it controls. Check/uncheck the checkbox to the left of a property to enable/disable editing of it. When you disable a property, that property will be entirely removed from the effect.

In this screen, you'll see all of the available parameters which can be set for the specific sub-effect. *Note for developers or advanced users: All of the available sub-effect properties are detailed in the reference for the various classes in the* `com.dwuser.ss4.pseudoStyleClasses.subEffectReferenceClasses` *package (in the API reference).* While in the wizard, hover over each property's name for a detailed description of what it controls:



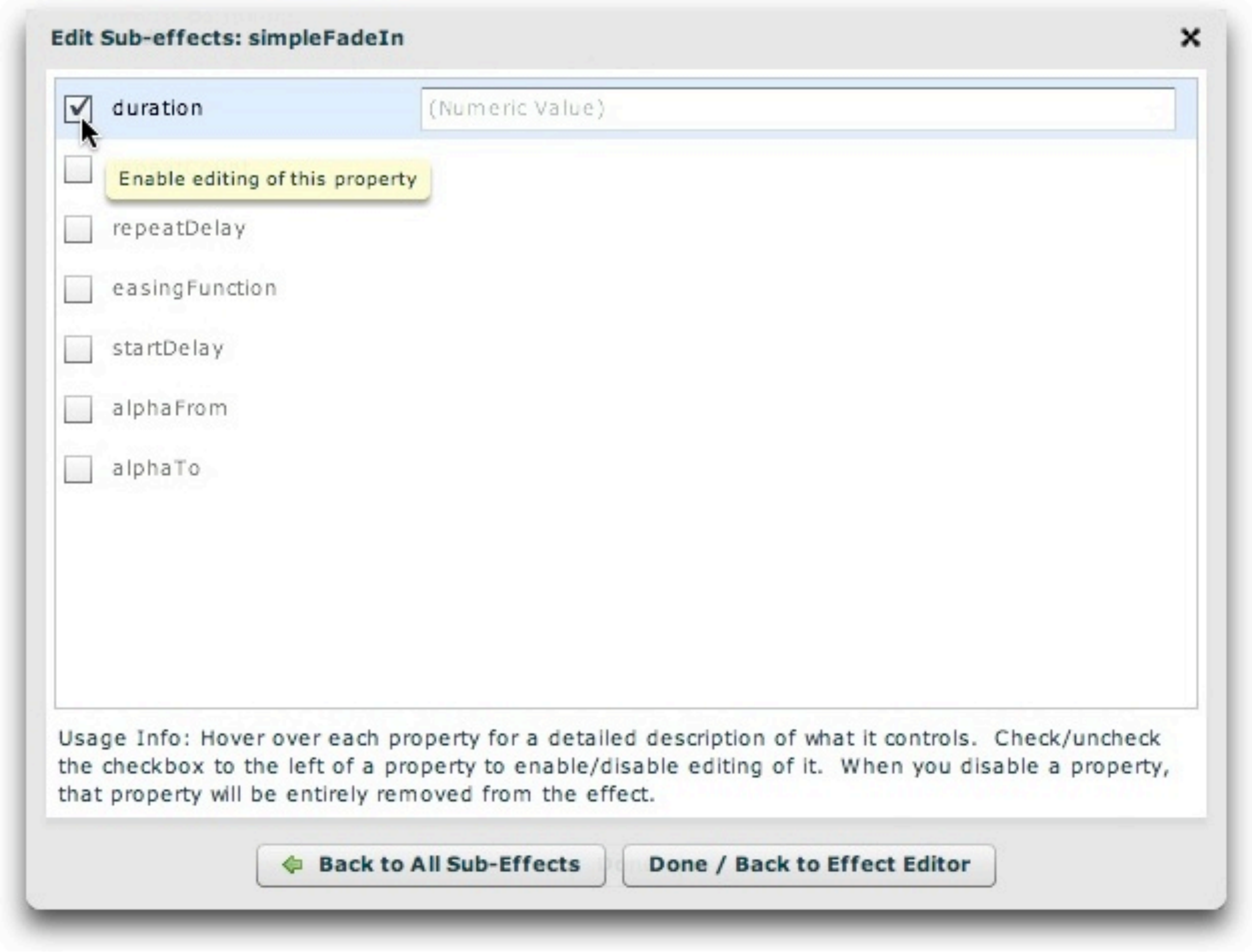
Format: Number

Duration of the effect in milliseconds.

For a repeated effect, the duration property specifies the duration of a single instance of the effect. Therefore, if an effect has a duration property set to 2000, and a repeatCount property set to 3, the effect takes a total of 6000 ms (6 seconds) to play.

Usage Info: Hover over each property for a detailed description of what it controls. Check/uncheck the checkbox to the left of a property to enable/disable editing of it. When you disable a property, that property will be entirely removed from the effect.

By default, none of the properties are enabled for editing; if you wish to specify a value for a property, check the checkbox to the left of the property name. Disabling the checkbox will remove that property from the sub-effect. The more commonly used properties are placed at the top of the list. For example, the **duration** specifies how long the effect lasts, and is the most important property.



Edit Sub-effects: simpleFadeIn

☑ duration (Numeric Value)

□

Enable editing of this property

□ repeatDelay

□ easingFunction

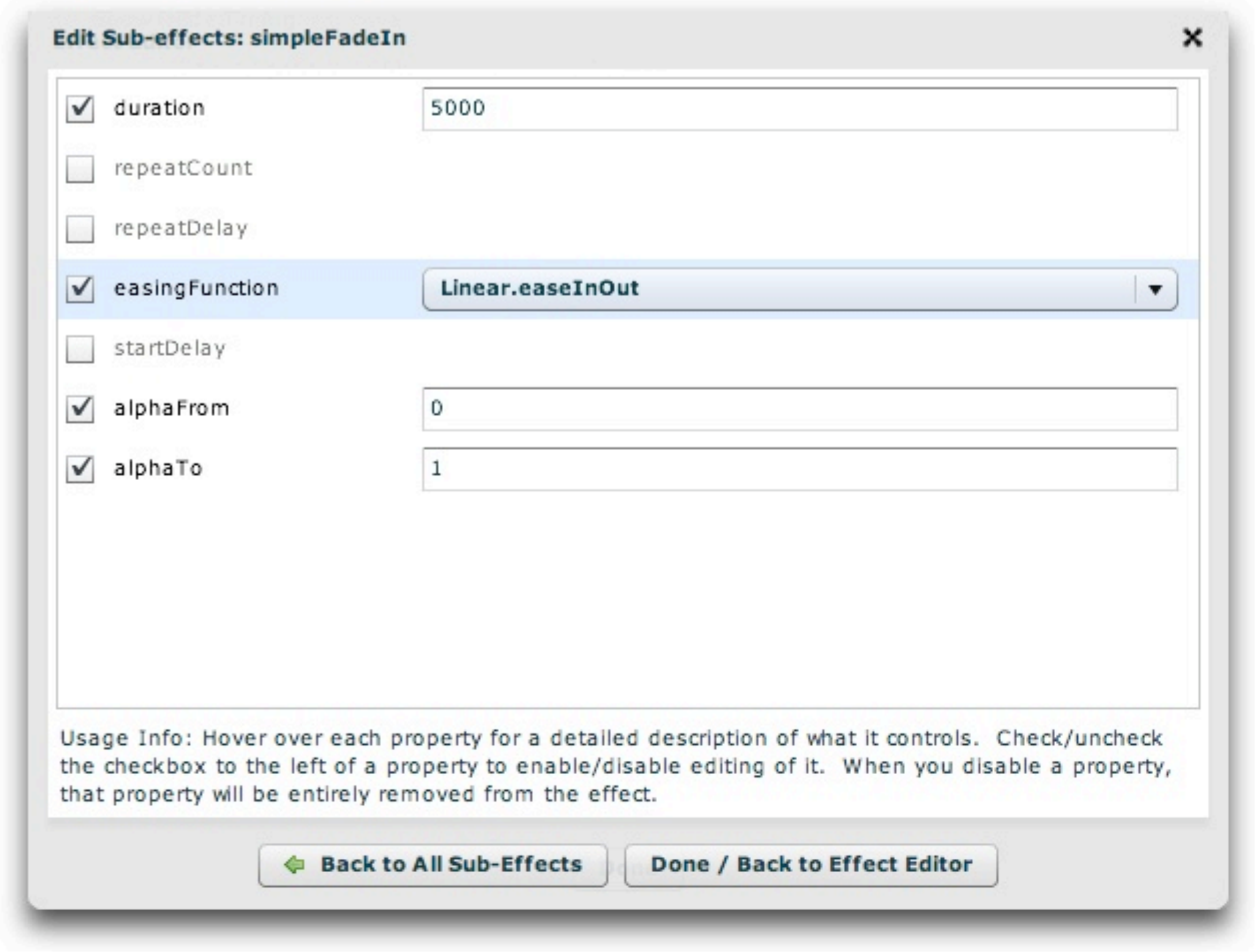
□ startDelay

□ alphaFrom

□ alphaTo

Usage Info: Hover over each property for a detailed description of what it controls. Check/uncheck the checkbox to the left of a property to enable/disable editing of it. When you disable a property, that property will be entirely removed from the effect.

For this example (a fade-in effect based on the Fade subeffect), the duration, alphaFrom, alphaTo, and easingFunction properties have been configured as follows:



Edit Sub-effects: simpleFadeIn

☑ duration 5000

□ repeatCount

□ repeatDelay

☑ easingFunction Linear.easeInOut

□ startDelay

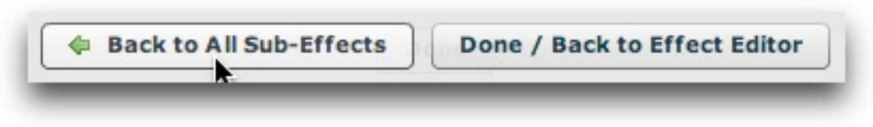
☑ alphaFrom 0

☑ alphaTo 1

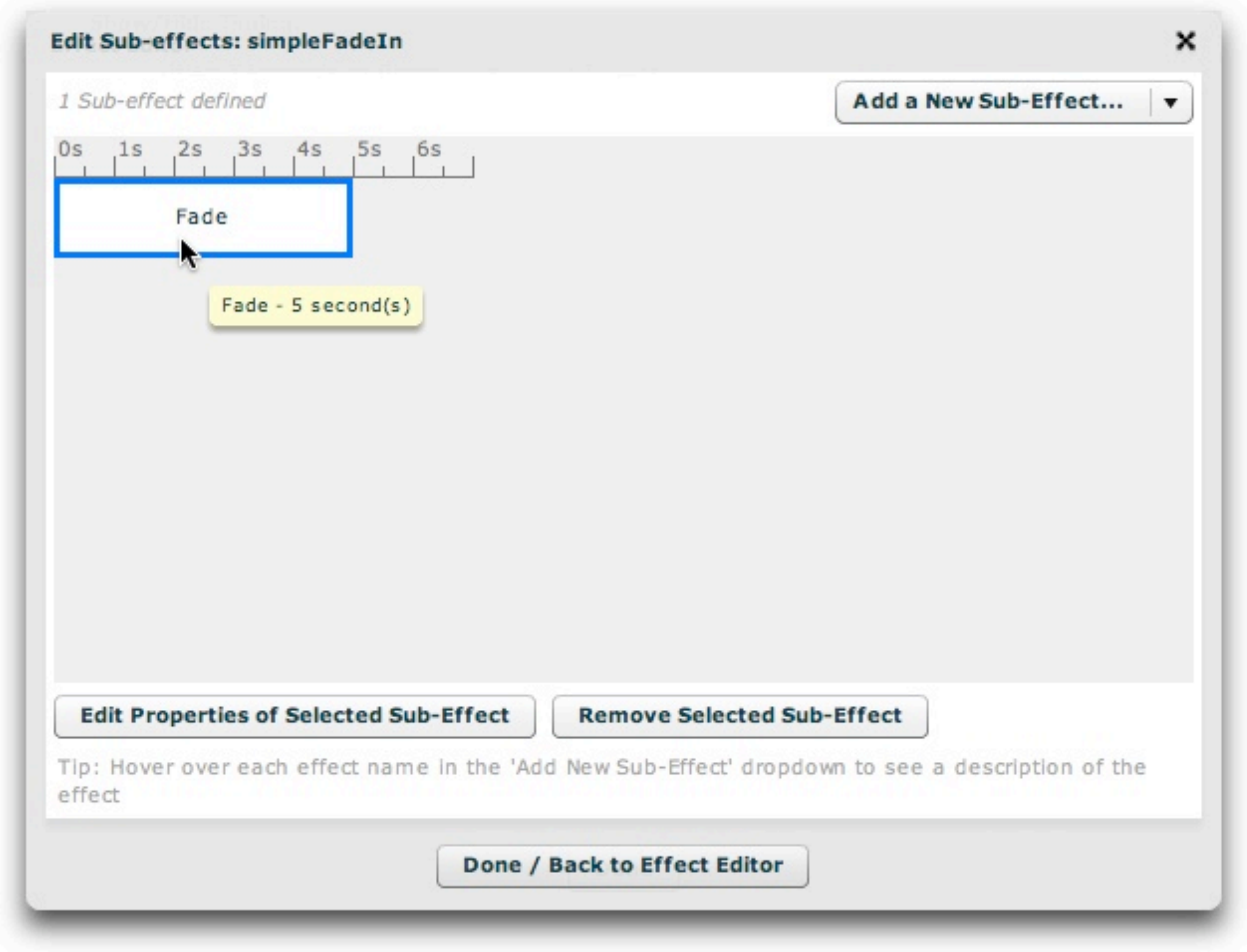
Usage Info: Hover over each property for a detailed description of what it controls. Check/uncheck the checkbox to the left of a property to enable/disable editing of it. When you disable a property, that property will be entirely removed from the effect.

Note: If you're setting a dropdown value but want to use the value which initially appears, it's a good idea to select two values to ensure that the property "sticks". In other words, in the screen above, you might select 'Linear.easeIn', then revert back to 'Linear.easeNone'. Otherwise, the wizard may not be able to know that you really wanted to set the value.

When you're done setting any properties, click the 'Back to All Sub-Effects' button to return to the sub-effect overview:



The newly added sub-effect will appear, represented by a selectable box:


Edit Sub-effects: simpleFadeIn

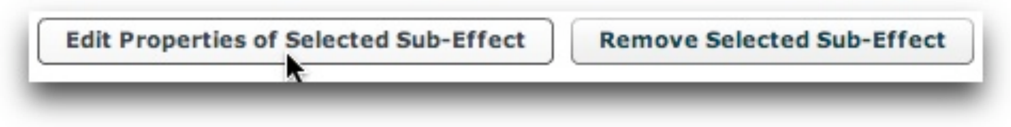
1 Sub-effect defined

Add a New Sub-Effect...
0s
1s
2s
3s
4s
5s
6s
Fade
Fade - 5 second(s)
Edit Properties of Selected Sub-Effect
Remove Selected Sub-Effect

Tip: Hover over each effect name in the 'Add New Sub-Effect' dropdown to see a description of the effect

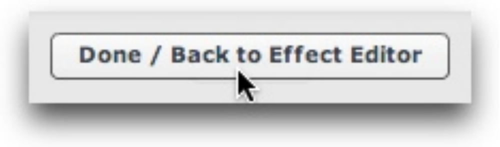
Done / Back to Effect Editor

Hovering over the box will display the name of the effect along with its duration. If you add other sub-effects, each sub-effect will have its own selectable box (examples of multiple sub-effects are below). To change any properties of a sub-effect, click on it to select it (its color will go from gray to white and blue), then click the 'Edit Properties of Selected Sub-Effect' button:


Edit Properties of Selected Sub-Effect
Remove Selected Sub-Effect

To remove the selected sub-effect, click the 'Remove Selected Sub-Effect' button.

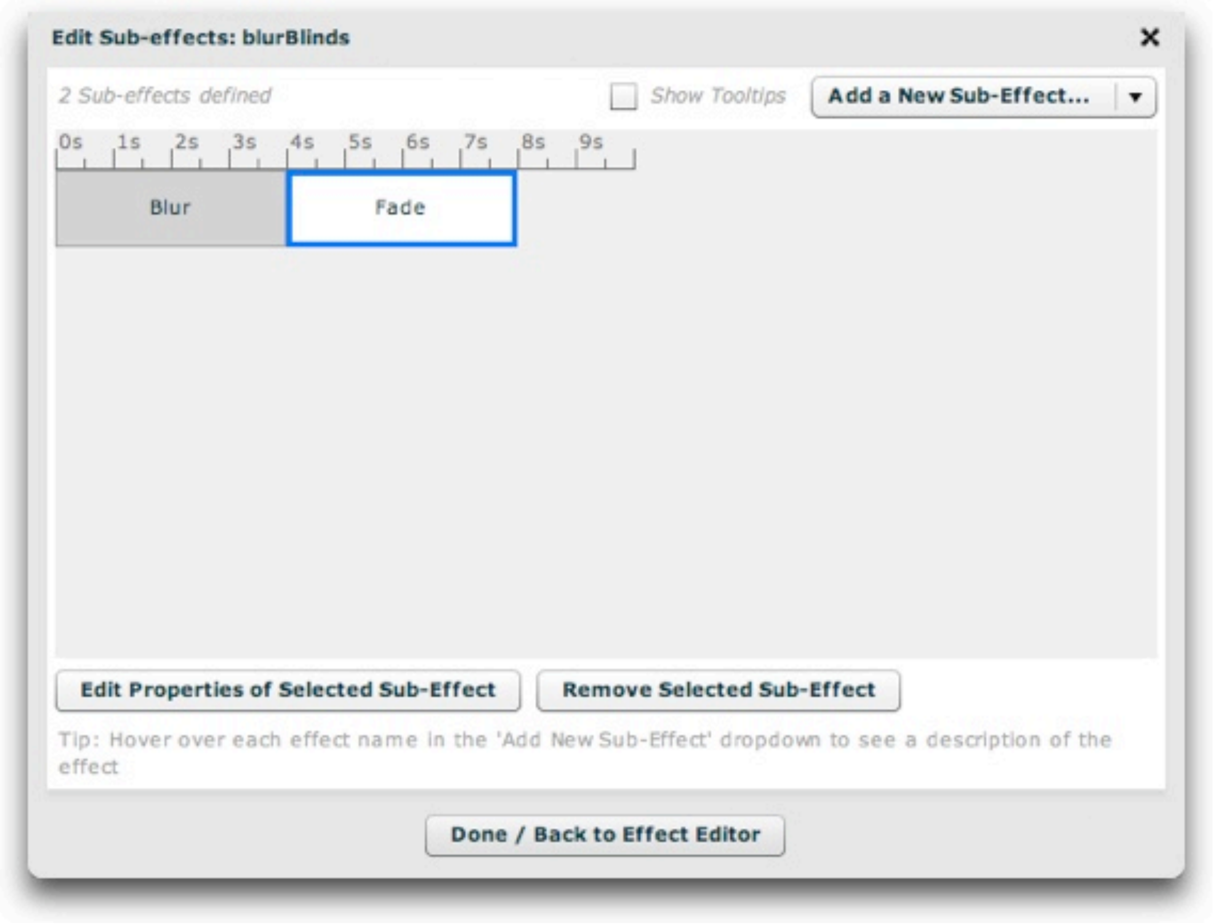
When you're done specifying the sub-effects for the effect, click the 'Done / Back to Effect Editor' to return to the main Effect Editor dialog:



Example: Using Multiple Sub-Effects

If you want to add another subeffect, follow the same process of using the 'Add a New Sub-Effect' menu again. Most effects only need one sub-effect, but sometimes you'll want to combine effects. For example, you might want to create an effect which both fades-out and blurs the target. To do this, you would add a Fade sub-effect and a Blur sub-effect.

Suppose that you wanted the blur effect to finish before the fade-out began. In this case, you would need to set the effect type to **Sequence**. The sub-effect representation would appear as follows:



Edit Sub-effects: blurBlinds

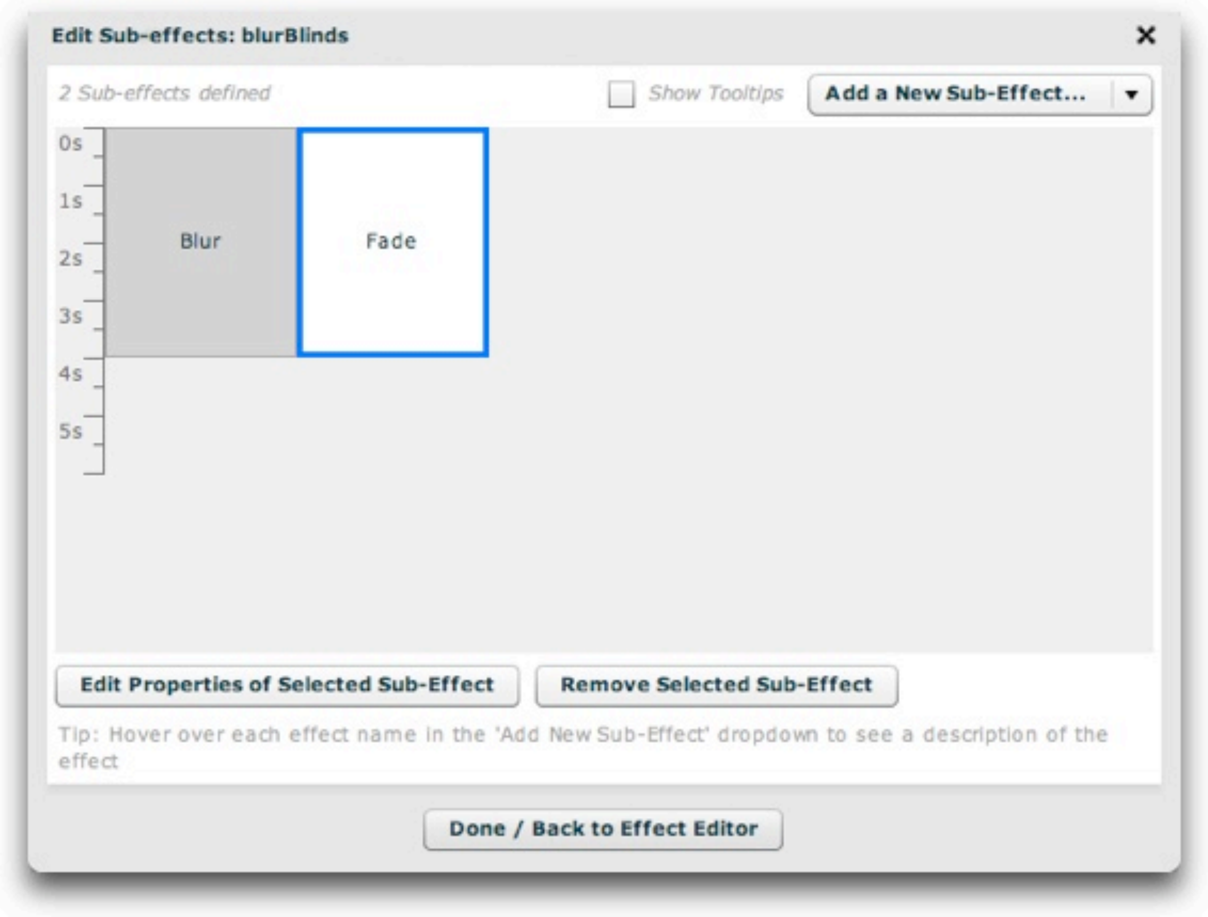
2 Sub-effects defined

□ *Show Tooltips*

Add a New Sub-Effect... ▾
0s
1s
2s
3s
4s
5s
6s
7s
8s
9s
Blur
Fade
Edit Properties of Selected Sub-Effect
Remove Selected Sub-Effect
Done / Back to Effect Editor

Tip: Hover over each effect name in the 'Add New Sub-Effect' dropdown to see a description of the effect

On the other hand, suppose that you wanted both effects to happen at the same time. In this case, you would need to set the effect type to **Parallel**. The sub-effect representation would appear as follows:


Edit Sub-effects: blurBlinds

2 Sub-effects defined

□ Show Tooltips

Add a New Sub-Effect...
0s
1s
2s
3s
4s
5s
Blur
Fade
Edit Properties of Selected Sub-Effect
Remove Selected Sub-Effect
Done / Back to Effect Editor

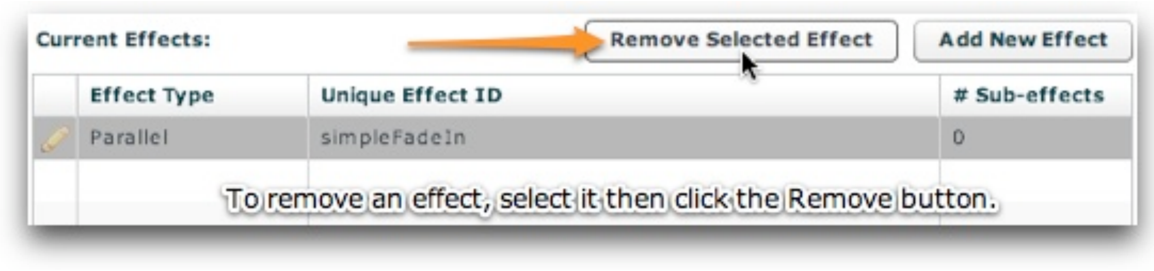
Tip: Hover over each effect name in the 'Add New Sub-Effect' dropdown to see a description of the effect

In both of these examples, the duration of both sub-effects was set to 4 seconds. This was done by setting the 'duration' property on both sub-effects to 4000 (since duration is specified in milliseconds, as noted in the property description).

Note: At this time, there is no built-in facility for re-ordering existing sub-effects. You can manually reorder them in the XML file (usually easiest), or recreate new sub-effects and remove the old ones which aren't in the desired order.

Removing an Effect

If you wish to remove an effect, select it in the grid then click the 'Remove Selected Effect' button:



	Effect Type	Unique Effect ID	# Sub-effects
✎	Parallel	simpleFadeIn	0

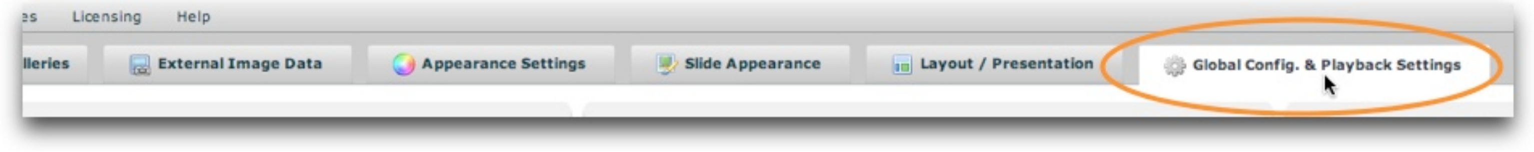
To remove an effect, select it then click the Remove button.

Important: If you have referenced an effect anywhere else in your slideshow, removing it will cause those linkages to fail. Remove effects with care.

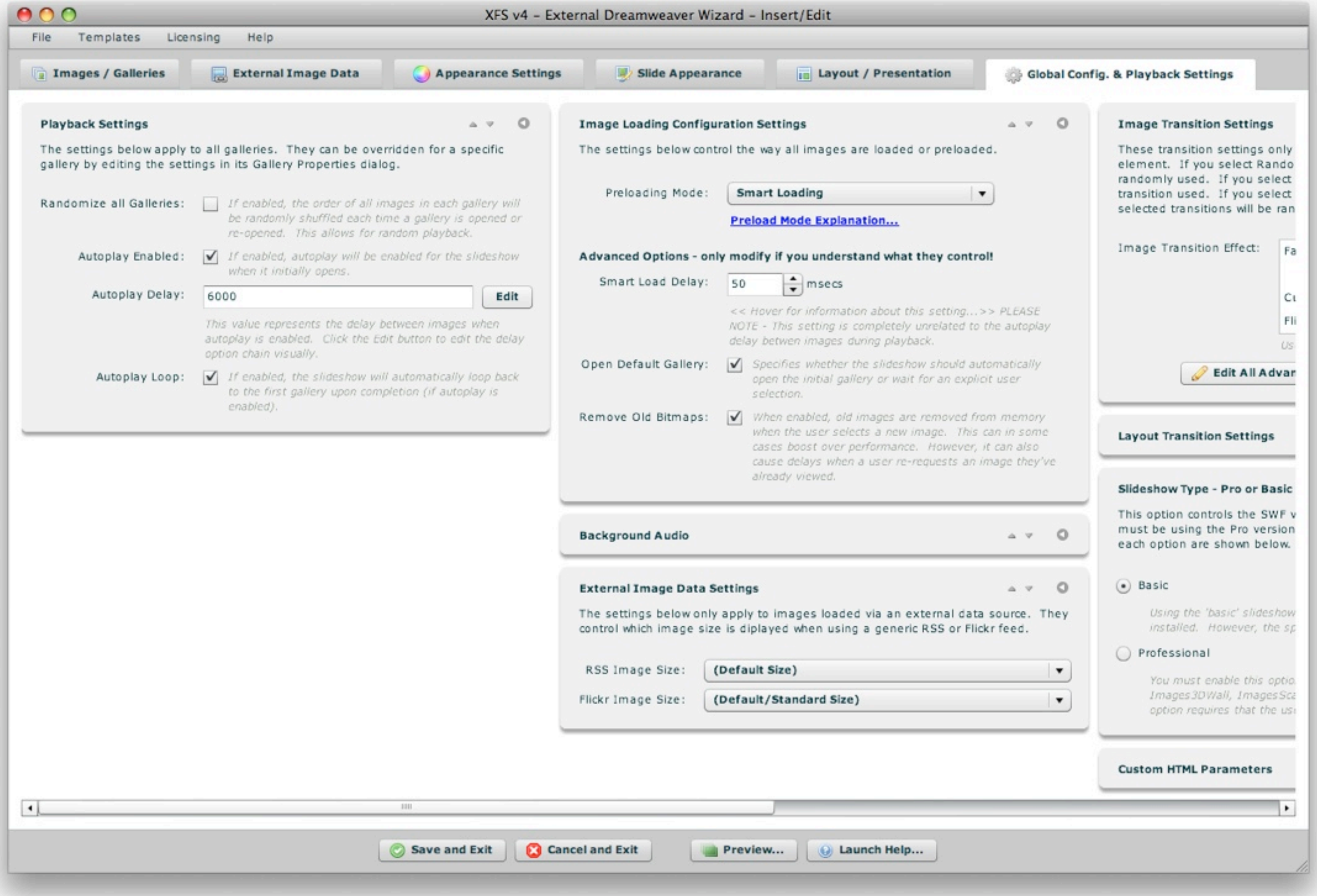
Using the External Wizard to Create and Edit Slideshows > Section-by-Section Usage Guide > **Using the 'Global Configuration and Playback Settings' Section**

Using the 'Global Configuration and Playback Settings' Section

The main Global Configuration and Playback Settings section is accessed by selecting the tab at the top of the wizard window:



The Global Configuration and Playback Settings section is where you will configure many settings which control the global configuration of the slideshow as well as the global playback configuration. These settings are all very important and should be reviewed carefully. The section appears as follows:



Playback Settings

The settings below apply to all galleries. They can be overridden for a specific gallery by editing the settings in its Gallery Properties dialog.

Randomize all Galleries: □ *If enabled, the order of all images in each gallery will be randomly shuffled each time a gallery is opened or re-opened. This allows for random playback.*

Autoplay Enabled: ☑ *If enabled, autoplay will be enabled for the slideshow when it initially opens.*

Autoplay Delay: 6000 Edit

This value represents the delay between images when autoplay is enabled. Click the Edit button to edit the delay option chain visually.

Autoplay Loop: ☑ *If enabled, the slideshow will automatically loop back to the first gallery upon completion (if autoplay is enabled).*

Image Loading Configuration Settings

The settings below control the way all images are loaded or preloaded.

Preloading Mode: Smart Loading

Preload Mode Explanation...

Advanced Options - only modify if you understand what they control!

Smart Load Delay: 50 msecs

<< Hover for information about this setting...>> PLEASE NOTE - This setting is completely unrelated to the autoplay delay betwen images during playback.

Open Default Gallery: ☑ *Specifies whether the slideshow should automatically open the initial gallery or wait for an explicit user selection.*

Remove Old Bitmaps: ☑ *When enabled, old images are removed from memory when the user selects a new image. This can in some cases boost over performance. However, it can also cause delays when a user re-requests an image they've already viewed.*

Background Audio

External Image Data Settings

The settings below only apply to images loaded via an external data source. They control which image size is diplayed when using a generic RSS or Flickr feed.

RSS Image Size: (Default Size)

Flickr Image Size: (Default/Standard Size)

Image Transition Settings

These transition settings only element. If you select Rando randomly used. If you select transition used. If you select selected transitions will be ran

Image Transition Effect: Fa Cu Fli

Us

Edit All Advar

Layout Transition Settings

Slideshow Type - Pro or Basic

This option controls the SWF v must be using the Pro version each option are shown below.

● Basic

Using the 'basic' slideshow installed. However, the sp

○ Professional

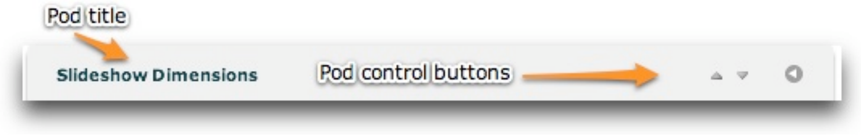
You must enable this optio Images3DWall, ImagesSca option requires that the us

Custom HTML Parameters

Navigating the Section / Using the Pods

The Global Configuration and Playback Settings panel is made up of a number of collapsible pods. You can expand the pods which are most relevant to you, and re-order them as desired. The order and expanded/collapsed state of the pods is automatically saved, and the last view is restored each time you re-launch the wizard. This is designed to streamline the development process, instantly placing the most relevant and commonly used settings at your fingertips.

Each pod has several control buttons at the top right:



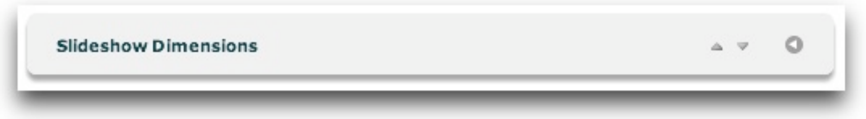
The first button allows you to move the pod up one space:

The second button allows you to move the pod down one space:

The third and final button allows you to expand or collapse the pod:

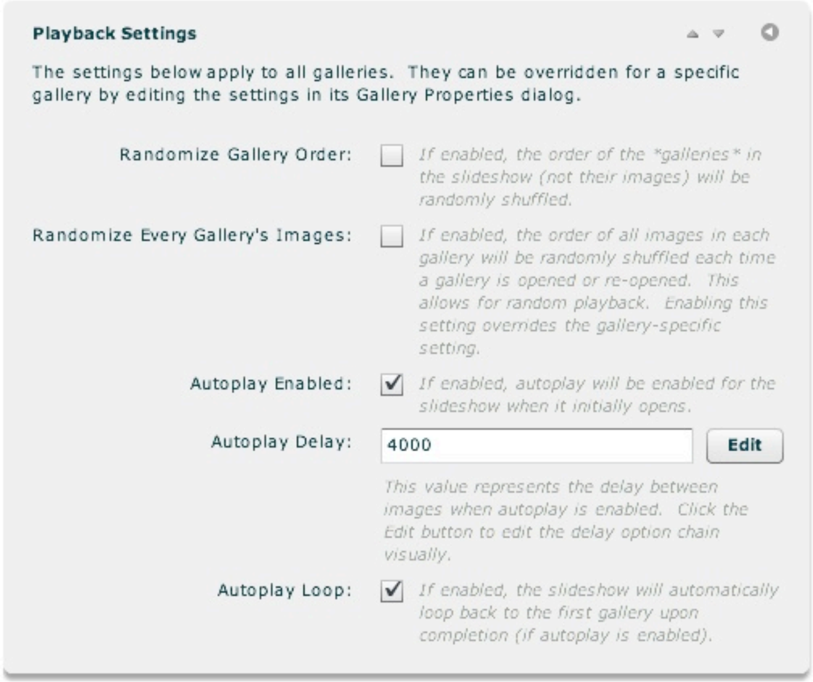
Tip: You can also expand or collapse a pod by clicking on its title.

When collapsed, a pod will appear like this:



Because there are so many settings in this section, you will likely need to use the horizontal scrollbar at the bottom of the screen to access all of the pods. The following sections of this topic detail each pod and the options each controls.

The Playback Settings Pod



Playback Settings

The settings below apply to all galleries. They can be overridden for a specific gallery by editing the settings in its Gallery Properties dialog.

Randomize Gallery Order: □ *If enabled, the order of the *galleries* in the slideshow (not their images) will be randomly shuffled.*

Randomize Every Gallery's Images: □ *If enabled, the order of all images in each gallery will be randomly shuffled each time a gallery is opened or re-opened. This allows for random playback. Enabling this setting overrides the gallery-specific setting.*

Autoplay Enabled: ☑ *If enabled, autoplay will be enabled for the slideshow when it initially opens.*

Autoplay Delay: 4000 Edit

This value represents the delay between images when autoplay is enabled. Click the Edit button to edit the delay option chain visually.

Autoplay Loop: ☑ *If enabled, the slideshow will automatically loop back to the first gallery upon completion (if autoplay is enabled).*

These settings determine the initial playback configuration for the entire slideshow as a whole.

Property	Description
Randomize Gallery Order	If enabled, the order of the galleries in the slideshow will be randomized. Note that this setting does not affect the order of the images in each gallery; only the gallery order is affected. The next setting must be used to randomize the images.
Randomize Every Gallery's Images	If enabled, the order of all images in each gallery will be randomized each time the gallery is opened. In other words, the order of all slides/images in the gallery is shuffled. This ensures true runtime randomization of the gallery every time it is viewed. If enabled, this setting overrides the gallery-specific setting in the Gallery Properties window.
Autoplay Enabled	Specifies if autoplay should be initially enabled for the entire slideshow.
Autoplay Delay	When autoplay is enabled, this value specifies the default delay between each image/slide. These values are specially constructed strings; clicking the Edit button allows you to visually edit the value. To learn more, see Using the Autoplay Delay Editor. This can be overridden for individual galleries or individual images.
Autoplay Loop	Specifies if the slideshow should loop back to the first image in the first gallery when the final image in the last gallery has been reached. If the Autoplay Loop setting is enabled for a single gallery, autoplay will loop within the gallery instead of skipping to the next gallery.

XML Setting Equivalents (for Advanced Users or Developers)

The Image Loading Configuration Settings Pod

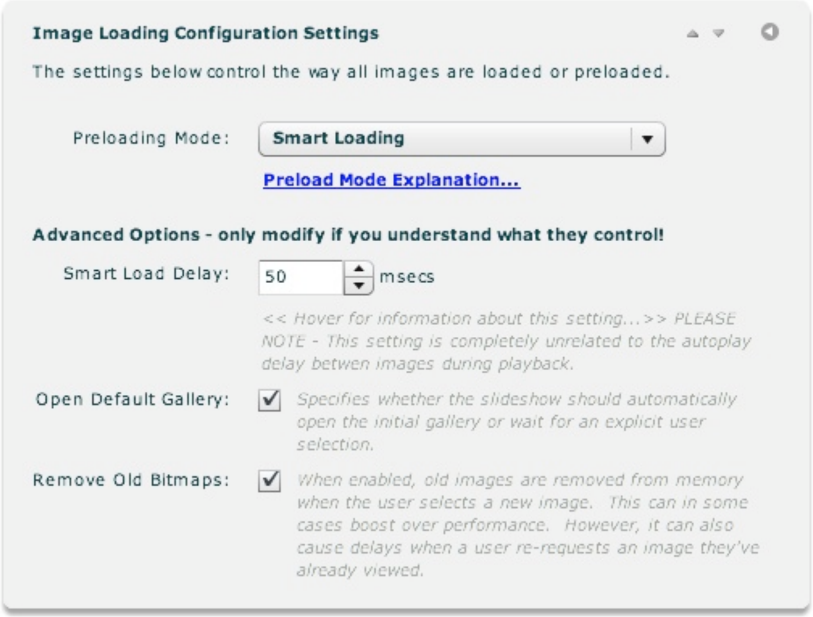


Image Loading Configuration Settings

The settings below control the way all images are loaded or preloaded.

Preloading Mode: Smart Loading ▾

Preload Mode Explanation...

Advanced Options - only modify if you understand what they control!

Smart Load Delay: 50 msecs

<< Hover for information about this setting...>> PLEASE NOTE - This setting is completely unrelated to the autoplay delay betwen images during playback.

Open Default Gallery: ☑ *Specifies whether the slideshow should automatically open the initial gallery or wait for an explicit user selection.*

Remove Old Bitmaps: ☑ *When enabled, old images are removed from memory when the user selects a new image. This can in some cases boost over performance. However, it can also cause delays when a user re-requests an image they've already viewed.*

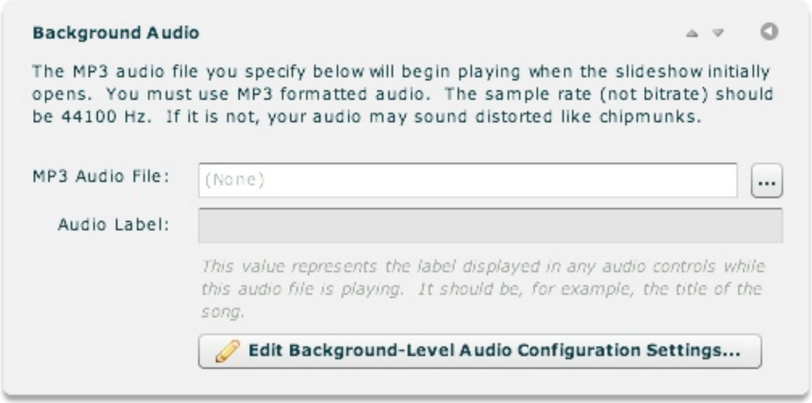
The settings in this pod control how images are loaded and managed.

Property	Description
Preloading Mode	Specifies how the images for each gallery are loaded. This setting will affect the loading time your users experience, and should be adjusted carefully. The available preloading modes are: ▪ **Smart Loading** - (`smart` XML attribute value) - This is usually the best option. As soon as the first image is loaded, it is displayed. However, in the background, loading of subsequent images continues. If the user chooses to skip to an unloaded image, that image is loaded and the loading manager then starts loading the images subsequent to it. ▪ **On-demand** - (`onDemand` XML attribute value) - This loading mode only loads each image when it is specifically requested. This option minimizes bandwidth usage, but can require your users to wait each time they request an image, especially if they have a slow connection or the images are very large. ▪ **Preload Entire Gallery** - (`preloadAll` XML attribute value) - Preloads all images in the gallery before the gallery begins to play. This loading mode is NOT recommended for most configurations, as it can cause lengthy up-front loading times. It is useful for certain timed presentations, however, when there must be an exact delay between every image (without any variations due to loading). ▪ **3D Display Preload** - (`3d` XML attribute value) - This preloading method should only be used for certain 3D display elements in the professional version. Oftentimes, the *Preload Thumbs* mode is a better choice. It preloads all of the thumbnail/low-res images before the gallery opens. When a specific image is requested, the low-res image is blown up to full size until the full-size high-res image can load. ▪ **Preload Thumbs, then Smart Loading** - (`preloadThumbs` XML attribute value) - This preloading mode preloads all thumbnail images before the gallery opens, then subsequently uses the Smart Loading method for all full-size images. This option should be used if it's important that the user have access to all thumbnails for navigation.

Smart Load Delay	This setting specifies the rest period, in milliseconds, between each image load when performing smart loading. This setting usually only makes a difference when a user re-visits a page and the slideshow images are already in the user's cache. In this case, it prevents the slideshow from bogging down as all of the images are loaded from the cache. PLEASE NOTE - This setting is completely unrelated to the autoplay delay betwen images during playback.
Open Default Gallery	Specifies whether the slideshow should automatically open the initial gallery or wait for an explicit user selection.
Remove Old Bitmaps	When enabled, old images are removed from memory when the user selects a new image. This can boost performance in some cases. However, it can also cause delays when a user re-requestes an image they've already viewed because it must be re-loaded into memory.

XML Setting Equivalents (for Advanced Users or Developers)

The Background Audio Pod



Background Audio

The MP3 audio file you specify below will begin playing when the slideshow initially opens. You must use MP3 formatted audio. The sample rate (not bitrate) should be 44100 Hz. If it is not, your audio may sound distorted like chipmunks.

MP3 Audio File: (None) ...

Audio Label:

This value represents the label displayed in any audio controls while this audio file is playing. It should be, for example, the title of the song.

Edit Background-Level Audio Configuration Settings...

The settings in this pod control the background audio (also known as Intro-level). Background audio is the audio which plays when a slideshow initially opens and is not associated with any specific gallery; it will play will all galleries are open. You can also configure it to only play prior to a gallery being selected in the Background-Level Audio Configuration Settings (detailed in the section below); this allows for separate intro and gallery-specific audio.

Property	Description
MP3 Audio File	Specifies the URL of the background audio MP3 file. The audio file must be in MP3 format; for necessary formatting guidelines, see Avoiding "Chipmunks" Audio: MP3 Formatting Guidelines. Enter the URL in the box or use the browse button to the right ('...') to select a file. The URL must be relative to the HTML page which holds the slideshow. **Note:** If you are loading audio from a **different domain** than the one on which the slideshow is displayed, you will be affected by Flash Player crossdomain loading security restrictions. To learn more, see this topic. **Note:** If you're audio plays back at a **distorted speed**, like "**chipmunks**", you need to review the MP3 formatting guidelines referenced above.

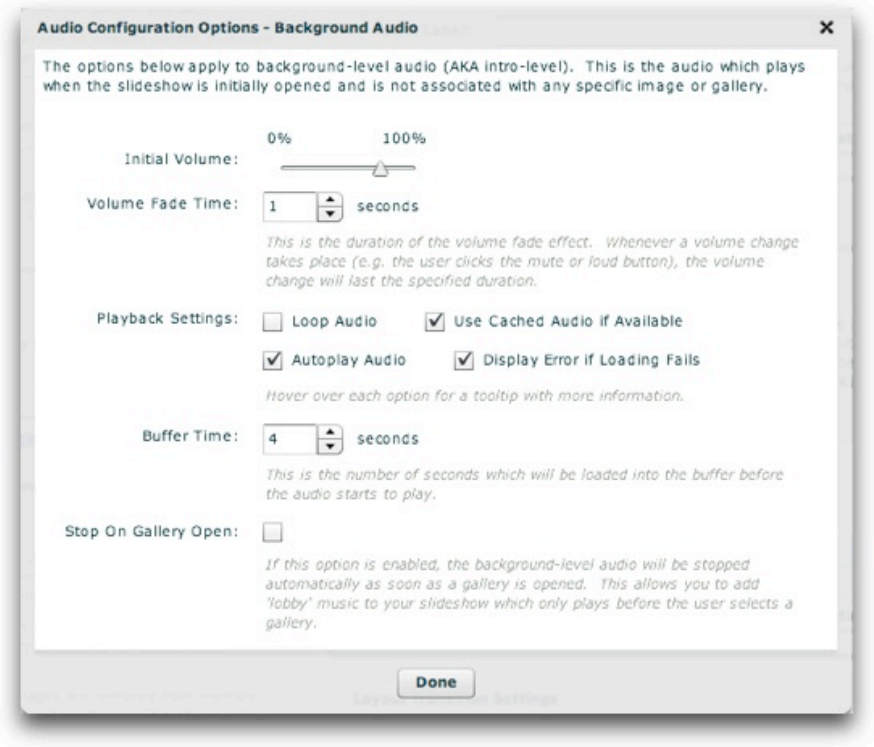
Audio Label	Once you specify an audio file, you can enter a string value in the Audio Label field. This value represents the label associated with the audio file; it is shown in any related audio controls while the file is playing. It should be the title of the song or the narration, for example.

XML Setting Equivalents (for Advanced Users or Developers)

The Background Audio Pod: Editing the Background-Level Audio Configuration Settings

There are several configuration settings, specific to background-level audio, which can be specified. These options are accessed through the 'Edit Background-Level Audio Configuration Settings...' button.

The dialog appears as follows. Note that the settings in this dialog apply to all image-level audio, not just the audio applied to a single image/slide.



Audio Configuration Options - Background Audio

The options below apply to background-level audio (AKA intro-level). This is the audio which plays when the slideshow is initially opened and is not associated with any specific image or gallery.

Initial Volume: 0% 100%

Volume Fade Time: 1 seconds

This is the duration of the volume fade effect. Whenever a volume change takes place (e.g. the user clicks the mute or loud button), the volume change will last the specified duration.

Playback Settings: □ Loop Audio ☑ Use Cached Audio if Available

☑ Autoplay Audio ☑ Display Error if Loading Fails

Hover over each option for a tooltip with more information.

Buffer Time: 4 seconds

This is the number of seconds which will be loaded into the buffer before the audio starts to play.

Stop On Gallery Open: □

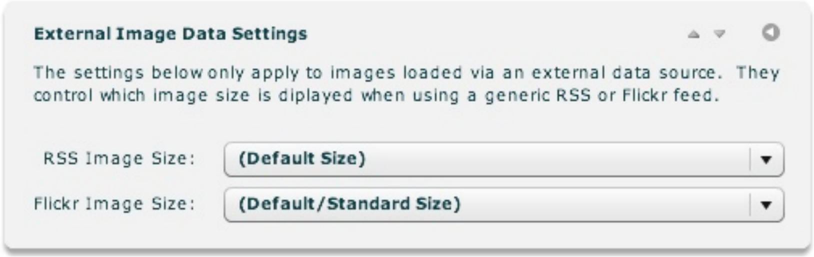
If this option is enabled, the background-level audio will be stopped automatically as soon as a gallery is opened. This allows you to add 'lobby' music to your slideshow which only plays before the user selects a gallery.

Each of the options controls the following (when working in the wizard, you can also learn more about some settings directly by hovering over their names):

Property	Description
Initial Volume	This is the initial volume of the audio file when it begins playing, excepting any user adjustments via a volume control. If you have provided user controls for volume, then this is simply the starting point. A value of 0% is entirely silent, whereas 100% is full volume.
Volume Fade Time	The volume fade effect fades the volume from its current value to the new value over the duration specified by this setting. Especially if you are using mute/loud volume controls (which jump directly to a certain volume), it is recommended that you utilize this effect. This provides a polished, less jarring transition when muting and unmuting audio.
Loop Audio	Specifies if the audio should loop back to the beginning when it finishes playing.
Use Cached Audio if Available	Rarely should this setting be disabled. If enabled, the slideshow will use a cached version of the audio file from the user's internet cache when available. This will improve loading performance if the user listens to the same file more than once, making it start almost instantly on subsequent loads. This setting should only be disabled in cases where the audio file is frequently updated.
Autoplay Audio	Specifies if the audio should start playing automatically. If this option is disabled, you must give the user audio controls and they must manually play the audio.
Display Error if Loading Fails	By default, an error will be shown when the loading of the audio fails. In a production setup, you may want to disable this setting. In rare instances on certain configurations, the audio may play correcly yet an error message appears. In these cases, the problem can be avoided by disabling the error reporting.
Buffer Time	This is the number of seconds of audio which must be loaded into the buffer before the audio starts to play. Additionally, if the buffer runs out, this duration will be loaded before playback resumes. Generally the default value of about 4 seconds is a good setting; it will ensure that even on slow connections the audio doesn't cut in and out.
Stop On Gallery Open	If this option is enabled, the background-level audio will be stopped automatically as soon as a gallery is opened. This allows the addition of 'lobby' music to your slideshow which only plays before the user selects a gallery.

XML Setting Equivalents (for Advanced Users or Developers)

The External Image Data Settings Pod



External Image Data Settings

The settings below only apply to images loaded via an external data source. They control which image size is diplayed when using a generic RSS or Flickr feed.

RSS Image Size: **(Default Size)**

Flickr Image Size: **(Default/Standard Size)**

When using external image data which includes a Media RSS feed, a Flickr feed, or Flickr API data, there may be multiple image sizes available. The settings in this pod allow you to choose the default sizes which will be used. For more information, see the Supported Dynamic Data Formats topic in the Using External Image Data section. For more Flickr-specific information, see 'Specifying the Flickr Image Sizes' within the Flickr Integration topic.

Property	Description
RSS Image Size	When using a generic Media RSS feed, specifies which image size should be used if multiple are available. Supported values include: ▪ **(Default Size)** - (`default` XML attribute value) - Specifies that the default image size should be used. In most Media RSS feeds with mulitiple sizes, the default will be specifically noted. ▪ **Largest Available** - (`largest` XML attribute value) - Specifies that the largest available image size should be used. ▪ **Smallest Available** - (`smallest` XML attribute value) - Specifies that the smallest available image size should be used.
Flickr Image Size	When using a Flickr feed or Flickr API data, there are multiple image sizes available to choose from. The size you choose will depend on the size of your slideshow. Supported values include: ▪ **Square Thumbnail** - (`square` XML attribute value) - Square Thumbnail Small square thumbnail, 75x75 pixels; image is cropped. ▪ **Small Thumbnail** - (`thumb` XML attribute value) - Small scaled version of original, 100 pixels on longest side; aspect ratio is maintained. ▪ **Medium** - (`medium` XML attribute value) - Medium-size scaled version of original, 240 pixels on longest side. ▪ **(Default/Standard size)** - (`standard` XML attribute value) - The default (standard) size; 500 pixels on longest side. ▪ **Big** - (`big` XML attribute value) - Large, 1024 pixels on longest side. Only exists for images with originals this large. If your source images aren't this large, loading will fail.

XML Setting Equivalents (for Advanced Users or Developers)

The Image Transition Settings Pod

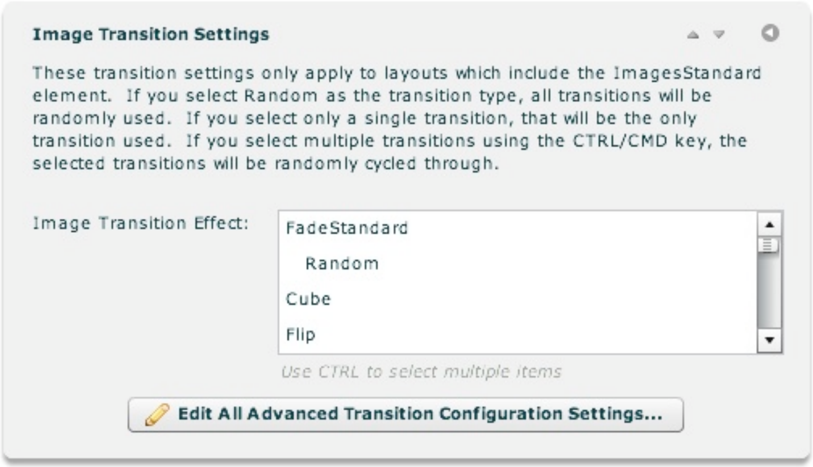


Image Transition Settings

These transition settings only apply to layouts which include the ImagesStandard element. If you select Random as the transition type, all transitions will be randomly used. If you select only a single transition, that will be the only transition used. If you select multiple transitions using the CTRL/CMD key, the selected transitions will be randomly cycled through.

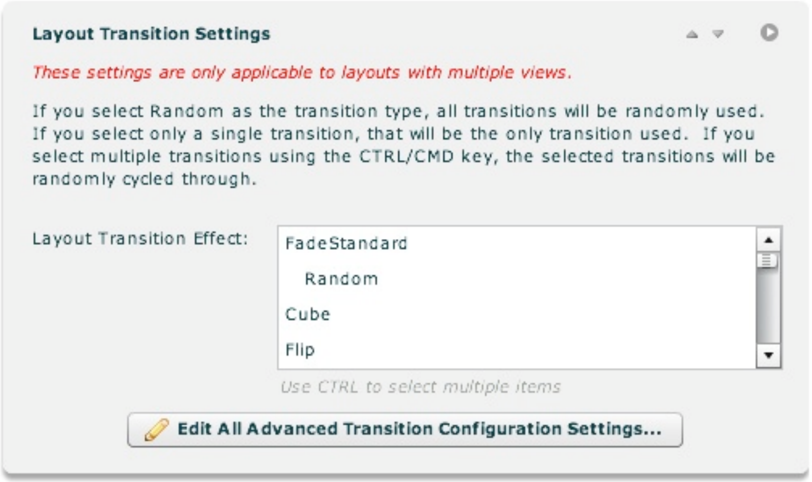
Image Transition Effect: FadeStandard

Random
Cube
Flip
Use CTRL to select multiple items
Edit All Advanced Transition Configuration Settings...

This pod only applies if you're using the ImagesStandard image display element (most slideshows do). In these cases, it allows you to specify the default image transition effect used between each slide / image. To learn more, see the Setting the Image Transition subtopic.

XML Setting Equivalents (for Advanced Users or Developers)

The Layout Transition Settings Pod



Layout Transition Settings

These settings are only applicable to layouts with multiple views.

If you select Random as the transition type, all transitions will be randomly used. If you select only a single transition, that will be the only transition used. If you select multiple transitions using the CTRL/CMD key, the selected transitions will be randomly cycled through.

Layout Transition Effect: FadeStandard / Random / Cube / Flip

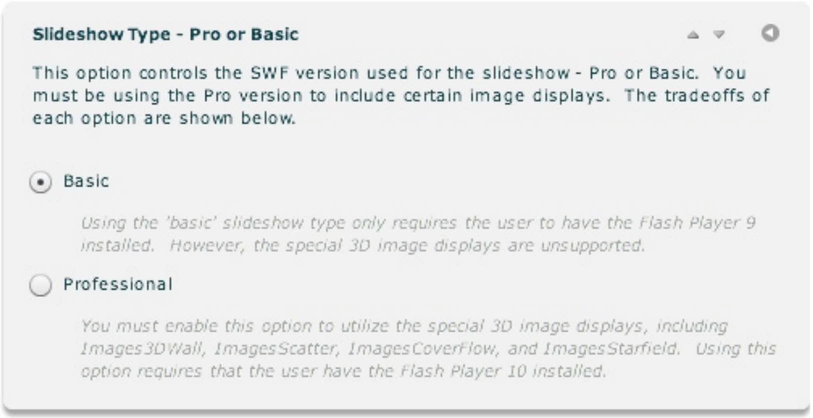
Use CTRL to select multiple items

The settings in this pod are only applicable when you have more defined multiple Normal layout views and have given the user controls to switch back and forth between these views (e.g. with the LayoutJumpButton element). In these cases, the settings control the transition effect between each layout view. The settings in this pod are identical to those in the Image Transition Settings pod; to learn how to further configure the layout transitions, see the Setting the Image Transition subtopic.

Note for Advanced Users and Developers: The only difference between the layout transition settings and the image transition settings is the style selector used. For the layout transitions, all of the wizard-based settings are used to set and modify properties on the `<LayoutTransition ... />` style selector node in the `<styles>` section of the XML configuration file.

XML Setting Equivalents (for Advanced Users or Developers)

The Slideshow Type - Pro or Basic Pod



Slideshow Type - Pro or Basic

This option controls the SWF version used for the slideshow - Pro or Basic. You must be using the Pro version to include certain image displays. The tradeoffs of each option are shown below.

● Basic

Using the 'basic' slideshow type only requires the user to have the Flash Player 9 installed. However, the special 3D image displays are unsupported.

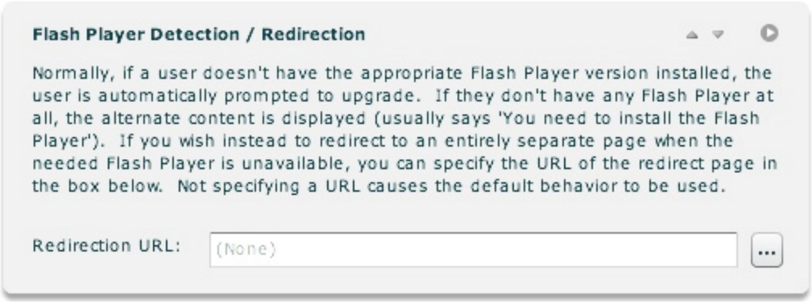
○ Professional

You must enable this option to utilize the special 3D image displays, including Images3DWall, ImagesScatter, ImagesCoverFlow, and ImagesStarfield. Using this option requires that the user have the Flash Player 10 installed.

The Slideshow Type setting specifies whether the Basic or Professional version SWF should be used. There are benefits to each version:

- **Basic** - Using the Basic SWF version only requires users to have the Flash Player 9 installed to view slideshows (98.9% penetration at writing). However, the special 3D image display elements are not supported.
- **Professional** - **Option available only in Professional version; to upgrade see here.** This option must be enabled to utilize the special 3D image display elements (Images3DWall, ImagesScatter, ImagesCoverFlow, and ImagesStarfield). It requires the user to have the Flash Player 10 installed to view slideshows (74.2% penetration at writing). Keep in mind that users are able to automatically upgrade if needed directly from your site using the built-in Express Install feature. This feature is also required if you want to force color correction to be enabled.

The Flash Player Detection / Redirection Pod



Flash Player Detection / Redirection

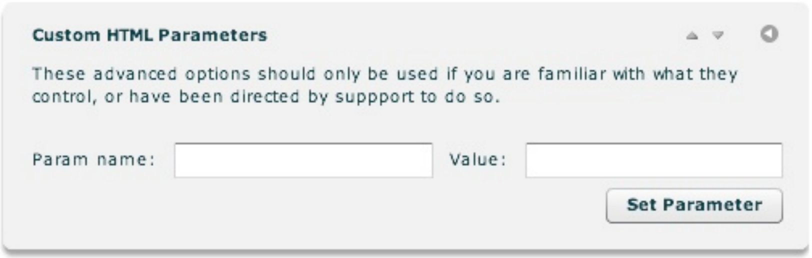
Normally, if a user doesn't have the appropriate Flash Player version installed, the user is automatically prompted to upgrade. If they don't have any Flash Player at all, the alternate content is displayed (usually says 'You need to install the Flash Player'). If you wish instead to redirect to an entirely separate page when the needed Flash Player is unavailable, you can specify the URL of the redirect page in the box below. Not specifying a URL causes the default behavior to be used.

Redirection URL: (None) ...

If you wish to have an entirely separate page for users who don't have the appropriate Flash Player, you can let them be automatically redirected by specifying a URL in this pod. If no redirection URL is specified, the user will instead be asked to upgrade or the alternate inline HTML content will be displayed.

Note for developers / advanced users: This parameter is equivalent to the `redirect` HTML parameter in the slideshow configuration object.

The Custom HTML Parameters Pod



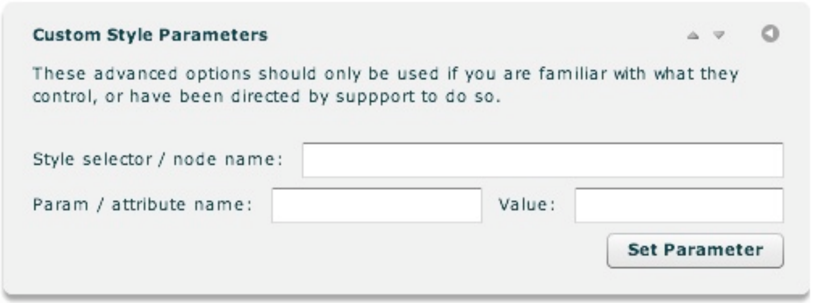
Custom HTML Parameters

These advanced options should only be used if you are familiar with what they control, or have been directed by suppport to do so.

This pod can be used by advanced users to manually specify slideshow HTML parameters (used for the initialization object or SWF embed settings). To learn more about all of the available initialization settings, see the `com.dwuser.ss4.core.FlashSlideshowConfigurationParameters` class in the API Reference. To explicitly specify a SWF embed parameter (e.g. `wmode`), use a parameter name in the format of `_param_wmode` -- in other words, just prepend `_param_` to the setting name.

Warning: Be careful using this section if you're not an advanced user; you can accidentally overwrite other settings.

The Custom Style Parameters Pod



Custom Style Parameters

These advanced options should only be used if you are familiar with what they control, or have been directed by suppport to do so.

This pod can be used to directly set XML style selector values. For an introduction to the concept and usage of global style selectors, see this topic on styling (a loose term, as these "styles" are also general configuration settings). The values set here are mapped to style setting nodes in the `<styles>` section of the XML configuration file.

For example, if you wanted to configure the global `IndividualSlideImage.@useSpinner` parameter so that a spinner loader is displayed on each slide, you would enter the following: `IndividualSlideImage` in the Style Selector field, `useSpinner` in the Param field, and `true` in the Value field.

For detailed reference on all available style selectors and their supported parameters, see the API Reference.

Warning: Be careful using this section if you're not an advanced user; you can accidentally overwrite other settings. Most settings are accessible directly through the wizard; it is best to use the standard wizard options when available.

Using the External Wizard to Create and Edit Slideshows > Section-by-Section Usage Guide > Using the 'Global Configuration and Playback Settings' Section > **Setting the Image Transition**

Setting the Image Transition

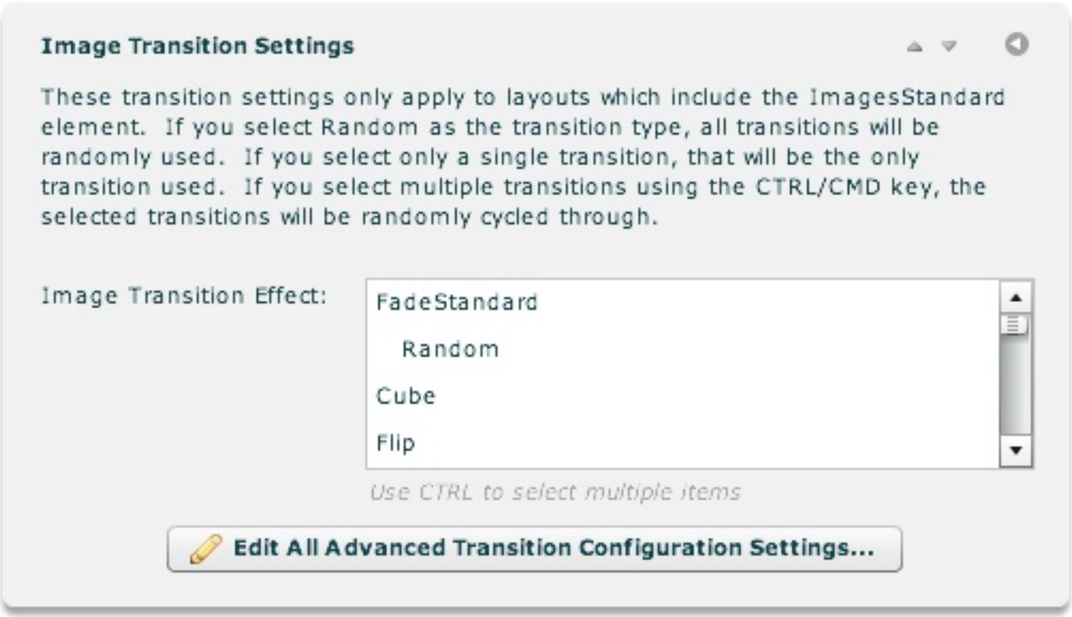


Image Transition Settings

These transition settings only apply to layouts which include the ImagesStandard element. If you select Random as the transition type, all transitions will be randomly used. If you select only a single transition, that will be the only transition used. If you select multiple transitions using the CTRL/CMD key, the selected transitions will be randomly cycled through.

Image Transition Effect:

Use CTRL to select multiple items

This pod only applies if you're using the ImagesStandard image display element (most slideshows do). In these cases, it allows you to specify the default image transition effect used between each slide / image. To view a description of each effect, you can hover over the effect name. Or, you can view the summary of all effects in the table below. If you don't wish to have the description tooltips, you can temporarily disable the tooltips by selecting File > Enable/Disable All Tooltips on the program menu.

Selecting multiple items (by holding down the CTRL or CMD key) will make the slideshow randomly cycle through those selected transitions. To cycle through *all* of the transitions randomly, select only the **Random** option.

If you do not specify an transition, the FadeStandard transition will be used by default.

To configure all of the available settings, see the Editing Transition Configuration Settings section below.

Description of Available Transitions

There are a number of image transitions which can be selected from. The following table summarizes them:

Transition Name	Description
`FadeStandard`	The FadeStandard transition is the standard fade transition used in most slideshows, where the old image fades out and the new image fades in (usually at the same time). This is the default transition.
`Random`	This option, if used, must be the sole selection. Selecting 'Random' means that **all** of the transitions will be randomly used when the images change. As mentioned above, you can also select several transitions and only these selected transitions will be randomly cycled through.
`Cube`	The Cube transition shows a cube which has the old and new images on adjacent faces. The cube rotates to transition from one image to the other.
`Flip`	The Flip transition is like a playing card with the old image on one face and the new image on the other. The card flips to transition from one image to the other.
`Pixelate`	The Pixelate transition makes the old image become highly pixelated and then become unpixelated. When it becomes unpixelated, the new image is revealed in the old image's place.
`Slide`	The Slide transition slides the old image out and slides the new image in at the same time, giving the appearance of both images being contiguous.
`BlurFadeFly`	The BlurFadeFly transition makes the old image blur, fade, and fly out. Then, the new image blurs, fades, and flies in.
`SoftWipe`	The SoftWipe transition makes the old image wipe away, while the new image wipes in. The wipe effect has a soft edge, giving a softened, polished look.
`PopInOut`	The PopInOut transition makes the old image 'pop' out of its place, then the new image 'pop' back in to replace it.
`PopIn`	The PopIn transition makes the new image 'pop' in on top of the old image, replacing it.
`PopOut`	The PopOut transition makes the old image 'pop' out to reveal the new image below it.
`BlurFadeInOut`	The BlurFadeInOut transition makes the old image blur and fade out, then the new image blurs and fades in to replace it.
`BlurFadeIn`	The BlurFadeIn transition makes the new image blur and fade in on top of the old image, replacing it.
`BlurFadeOut`	The BlurFadeOut transition makes the old image blur and fade out to reveal the new image below it.
`PixelDissolveInOut`	The PixelDissolveInOut transition uses the PixelDissolve effect to hide the old image block-by-block, then uses the same type of effect to reveal the new image.
`PixelDissolveIn`	The PixelDissolveIn transition uses the PixelDissolve effect to reveal the new image on top of the old image, replacing it.

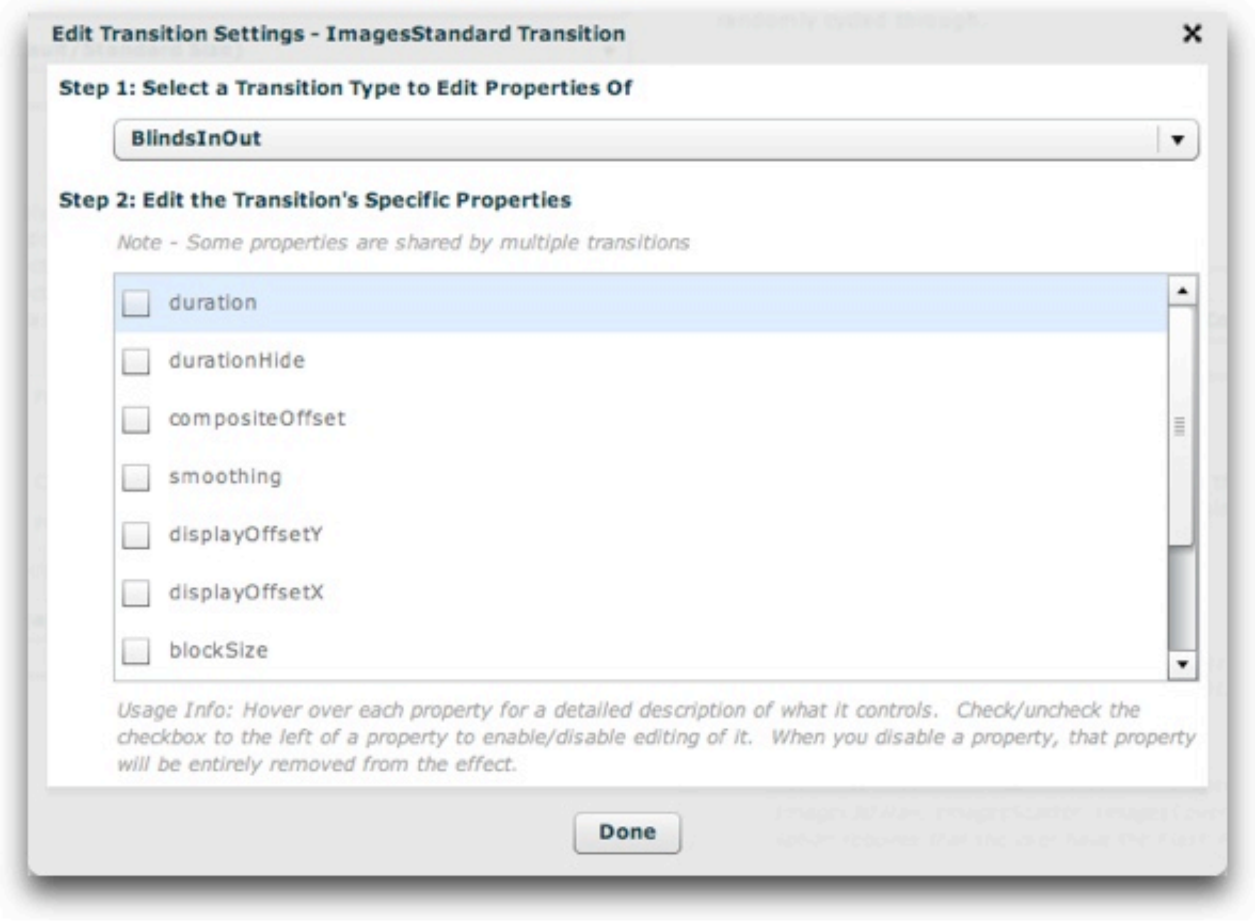
`PixelDissolveOut`	The PixelDissolveOut transition uses the PixelDissolve effect to hide the old image block-by-block, revealing the new image below it.
`SoftIrisInOut`	The SoftIrisInOut transition uses an Iris effect to hide the old image, then uses the same type of effect to reveal the new image.
`SoftIrisIn`	The SoftIrisIn transition uses an Iris effect to reveal the new image on top of the old image, replacing it.
`SoftIrisOut`	The SoftIrisOut transition uses an Iris effect to hide the old image, revealing the new image below it.
`BlindsInOut`	The BlindsInOut transition uses a Blinds effect to hide the old image, then uses the same type of effect to reveal the new image.
`BlindsIn`	The BlindsIn transition uses a Blinds effect to reveal the new image on top of the old image, replacing it.
`BlindsOut`	The BlindsOut transition uses a Blinds effect to hide the old image, revealing the new image below it.

Editing Transition Configuration Settings

There are a number of different settings which affect the appearance, duration, etc. of the image transitions(s) used. Some of these parameters are unique to a given effect; others are shared by many or all transitions. To customize these settings, click the 'Edit All Advanced Transition Configuration Settings...' button:

Tip for Advanced Users and Developers: To learn in more detail about all of the available attributes (properties) for each transition, see the `com.dwuser.ss4.ui.ImagesStandardTransition` class documentation in the API Reference. All of the wizard-based settings shown below are used to set and modify properties on the `<ImagesStandardTransition ... />` style selector node in the `<styles>` section of the XML configuration file.

The Edit Transition Settings dialog will open:



Edit Transition Settings - ImagesStandard Transition

Step 1: Select a Transition Type to Edit Properties Of

BlindsInOut

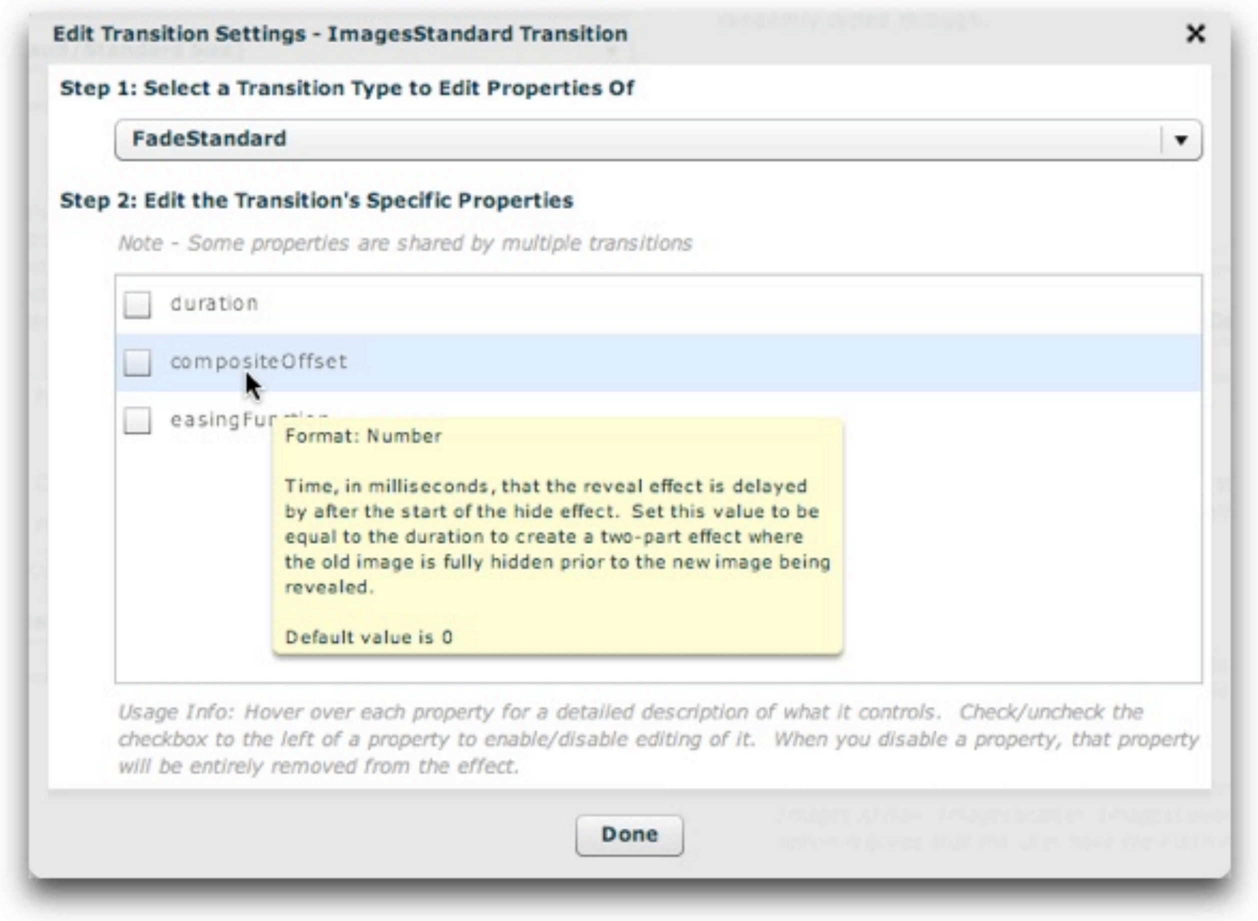
Step 2: Edit the Transition's Specific Properties

Note - Some properties are shared by multiple transitions

- □ duration
- □ durationHide
- □ compositeOffset
- □ smoothing
- □ displayOffsetY
- □ displayOffsetX
- □ blockSize

Usage Info: Hover over each property for a detailed description of what it controls. Check/uncheck the checkbox to the left of a property to enable/disable editing of it. When you disable a property, that property will be entirely removed from the effect.

First, you must select the transition for which to edit the properties. You will want to select each of the transitions you plan on using in the slideshow. Once a selection has been made in the 'Step 1' dropdown, you'll see all of the available parameters which can be set for that specific transition. *(Advanced Users and Developers: See the above note for detailed documentation on available properties.)* While in the wizard, hover over each property's name for a detailed description of what it controls along with the property's default value for that transition:



Edit Transition Settings - ImagesStandard Transition

Step 1: Select a Transition Type to Edit Properties Of

FadeStandard

Step 2: Edit the Transition's Specific Properties

Note - Some properties are shared by multiple transitions

□ duration

□ compositeOffset

□ easingFun

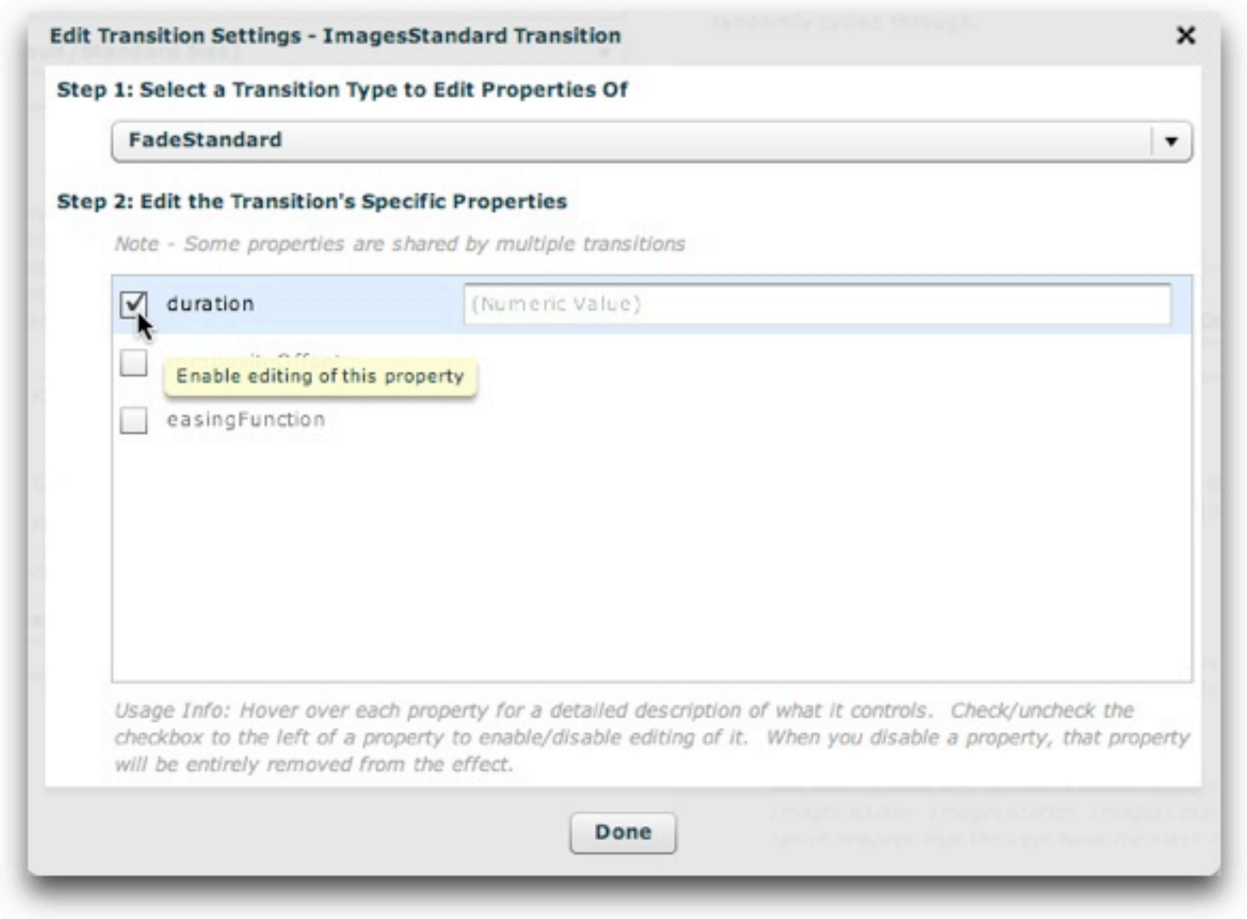
Format: Number

Time, in milliseconds, that the reveal effect is delayed by after the start of the hide effect. Set this value to be equal to the duration to create a two-part effect where the old image is fully hidden prior to the new image being revealed.

Default value is 0

Usage Info: Hover over each property for a detailed description of what it controls. Check/uncheck the checkbox to the left of a property to enable/disable editing of it. When you disable a property, that property will be entirely removed from the effect.

By default, none of the properties are enabled for editing; if you wish to specify a value for a property, check the checkbox to the left of the property name. Disabling the checkbox will remove that property from the transition settings. The more commonly used properties are placed at the top of the list. For example, the **duration** specifies how long the effect lasts, and is the most important property.



Edit Transition Settings - ImagesStandard Transition

Step 1: Select a Transition Type to Edit Properties Of

FadeStandard ▾

Step 2: Edit the Transition's Specific Properties

Note - Some properties are shared by multiple transitions

☑ duration (Numeric Value)

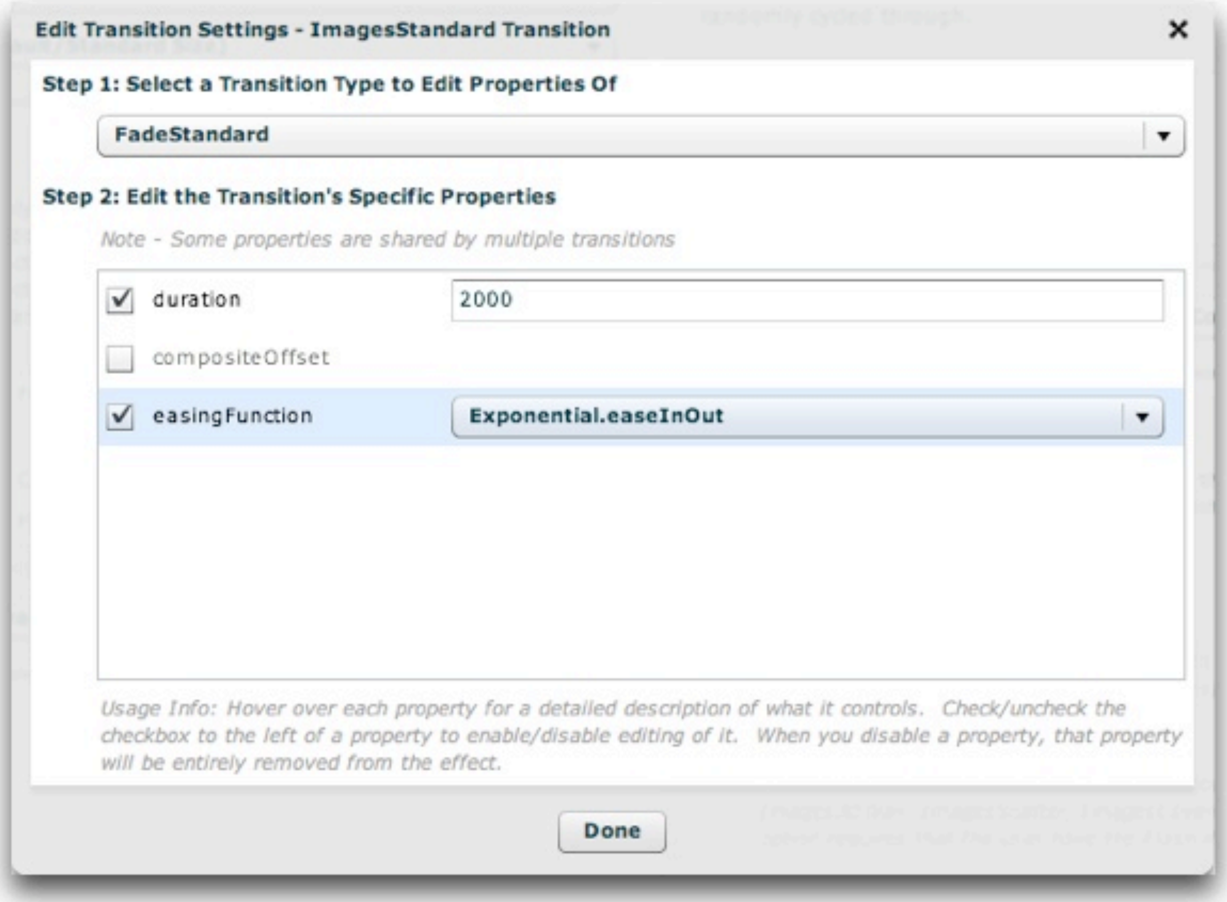
□

Enable editing of this property

□ easingFunction

Usage Info: Hover over each property for a detailed description of what it controls. Check/uncheck the checkbox to the left of a property to enable/disable editing of it. When you disable a property, that property will be entirely removed from the effect.

For this example, the `duration` and `easingFunction` properties have been configured as follows for the FadeStandard transition:



Edit Transition Settings - ImagesStandard Transition

Step 1: Select a Transition Type to Edit Properties Of

FadeStandard

Step 2: Edit the Transition's Specific Properties

Note - Some properties are shared by multiple transitions

☑ duration 2000

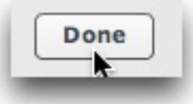
□ compositeOffset

☑ easingFunction Exponential.easeInOut

Usage Info: Hover over each property for a detailed description of what it controls. Check/uncheck the checkbox to the left of a property to enable/disable editing of it. When you disable a property, that property will be entirely removed from the effect.

Note: If you're setting a dropdown value but want to use the value which initially appears, it's a good idea to select two values to ensure that the property "sticks". In other words, in the screen above, you might select 'Linear.easeIn', then revert back to 'Linear.easeNone'. Otherwise, the wizard may not be able to know that you really wanted to set the value.

When you're done setting any properties, click the 'Done' button to exit:



Example: Creating a Separate Fade-In / Fade-Out Transition

By default, most transitions hide the old image and reveal the new image at the same time. If you wish for both to take place separately, use the Composite Offset setting in the Edit Transition Configuration Settings dialog. For example, if you set Duration to 2000, set Composite Offset to 2000 as well. This would make the old image fade out over two seconds, then have the new image immediately fade in over two seconds.

Developer Reference

This section of the manual provides detailed information about the HTML and XML structure behind the GUI. It's the "man behind the curtain" of the XFS v4.

Important Notes

Before beginning, a disclaimer -- this section is not for beginners or novices. If you're not familiar with code editing or aren't sure what XML means, you need to stick with the other sections of the manual.

The developer reference is essentially a description of the XML framework used to hold all of the actual data. For many aspects, the complete in-depth reference can be found in the API Reference. Links will be provided where appropriate. Throughout the rest of the documentation, most settings reference the corresponding XML settings. So, if you're a developer, don't assume that you can skip over the other sections; they are complementary to the contents here.

When editing directly, it is still necessary to use the Dreamweaver wizard to initially insert the slideshow. You don't have to add any data; just run the wizard, Save, Close, and Insert. This will generate the file structure which is needed for the slideshow to appear properly and copy all supporting files into place.

General Tips

When editing XML, you can accidentally make typos which create syntax errors and make the XML document unable to be parsed. If you make such a mistake, you will be notified that parsing failed when you attempt to view the slideshow. The error sometimes isn't very precise; to determine the precise location of an error, open the XML document in Firefox. Firefox offers excellent XML display as well as detailed error location notification.

Developer Reference > **Overview of Slideshow HTML structure**

Overview of Slideshow HTML Structure

When you insert a slideshow into an HTML page, a block of code referred to as 'Slideshow HTML' is inserted. It begins and ends as follows:

```
<!-- Start dwuser_XML_Flash_Slideshow_v4 -->
...Slideshow code here...
<!-- End dwuser_XML_Flash_Slideshow_v4 -->
```

There are three main components to each of these blocks:

The Supporting JS File Reference

First is the `<script>` reference to the `slideshow.js` embed script in the `v4flashslideshow/` supporting files folder. It appears as follows in the Slideshow HTML:

```
<!-- Do not remove the line below!!! It is required for the DWUser XML Flash Slideshow v4. -->
<script type="text/javascript" src="v4flashslideshow/slideshow.js"></script>
```

When you initially insert a slideshow, the bolded reference will automatically be generated so that it correctly points at the `slideshow.js` file. It is important to make sure that this reference is updated if you change any file locations. If you ever accidentally break the reference (or forget to upload the supporting files), an alert box will be displayed when you attempt to view the slideshow in the browser, noting the problem.

The Alternate Content Holder

Next is the `<div>` tag which will be replaced by the slideshow in your page. This `<div>` tag holds all of the alternate content which is displayed if the Flash Player or Javascript are unavailable:

```
<div class="dwuser_xfs_v4_holder" style="width: 600px; height: 400px;">
     <strong><a href="http://www.adobe.com/go/getflashplayer/">You need to upgrade your Flash Player and
```

```
enable Javascript to view this content &raquo;</a></strong>
</div>
```

To learn more about alternate content and how to customize it, see the Alternate HTML Content subtopic.

The Slideshow HTML Parameters / Embed Call

Finally is the slideshow embed function call. All of the slideshow initialization parameters, including the SWF embed parameters, are passed in a javascript object to this function:

```
XMLFlashSlideshow_v4({width:'600', height:'400', xml:'v4flashslideshow/slideshow_data0.xml',
backgroundColor:'#C5C5C5', backgroundAlpha:'0.5', preloaderTextColor:'#000000'});
```

These parameters pass important data, most especially the path to the XML configuration file (bolded above). To learn more about the slideshow embed call, see the Slideshow HTML Parameters subtopic.

Alternate HTML Content

When a slideshow is embedded in your HTML page (this is done using SWFObject), it actually overwrites a block of regular HTML after confirming that the appropriate Flash Player version is available. If the appropriate Flash Player version is unavailable but the user does have the Flash Player 6.0.65 or higher installed, they will automatically be prompted to use Express Install to upgrade their Flash Player directly from your site. Once this completes, the page will reload and the slideshow will display properly. If, however, the user has disabled Javascript or he/she does not have the Flash Player 6.0.65 or higher, the embedding process will not be able to take place. In this case, the alternate content in the slideshow holder `<div>` tag (in your Slideshow HTML) will be displayed instead. The HTML in this block determines what the user sees when the slideshow is unable to be embedded. Here is the default holder `<div>`:

```
<div class="dwuser_xfs_v4_holder" style="width: 600px; height: 400px;">
     <strong><a href="http://www.adobe.com/go/getflashplayer/">You need to upgrade your Flash Player and
enable Javascript to view this content &raquo;</a></strong>
</div>
```

You can include an image here if such is an acceptable substitution. If you definitely want your users to be able to view the slideshow, you should display a message with two components:

- Have the user make sure they have Javascript enabled
- Have the user upgrade their Flash Player - link to `http://get.adobe.com/flashplayer/` or `http://www.adobe.com/go/getflashplayer/` .

You can display a message tailored to each case by using the following code, for example:

```
<div class="dwuser_xfs_v4_holder" style="width: 600px; height: 400px;">
   <noscript>You need to enable Javascript to view this slideshow.</noscript>
   <script language="javascript">
       document.write('<strong><a href="http://www.adobe.com/go/getflashplayer/">You need to upgrade your
Flash Player to view this slideshow &raquo;</a></strong> ');
   </script>
</div>
```

The listed code will display an 'Enable Javascript' message if javascript is disabled, and an 'Upgrade your Flash Player' message if the issue is that they have a very old Flash Player (or none at all).   Here is an example of using an image as alternate content:

```
<div class="dwuser_xfs_v4_holder" style="width: 600px; height: 400px;">
     <img src="path/to/slideshow_image.jpg" width="600" height="400" />
</div>
```

Don't spend too much time on alternate content -- generally, only about 1 percent of your visitors will not have the required settings / plugins in view the slideshow, and they'll receive upgrade messages on almost every site they visit.

Slideshow HTML Parameters

The slideshow configuration object is passed to the slideshow HTML through the `XMLFlashSlideshow_v4(...)` embed function call. This object is a javascript object which has a number of name/value pairs. Here is an example call:

```
XMLFlashSlideshow_v4({width:'600', height:'400', xml:'v4flashslideshow/slideshow_data0.xml',
backgroundColor:'#C5C5C5', backgroundAlpha:'0.5', preloaderTextColor:'#000000'});
```

All parameters are separated by commas and are in the format `parameterName:'value'` .

The most important parameter is the `xml` parameter, which passes the path to the XML configuration file as well as any other external image data sources (with each separated by a comma). You can learn more about the format used for linking external data sources to specific galleries in the Using External Image Data section of the manual; see Understanding How Dynamic Data Works and Supported Dynamic Data Formats for helpful information.

Also passed are a number of settings which control the appearance of the slideshow before the configuration data is loaded. These include the width, height, background appearance, and preloader apperance.

Only a few parameters are usually necessary (such as those shown in the example above); only `width`, `height`, and `xml` are required. All of available parameters are detailed in the `com.dwuser.ss4.core.FlashSlideshowConfigurationParameters` class in the API Reference.

In the wizard, most of these settings are specified in the Apperance Settings section, and to a lesser degree the Global Configuration and Playback Settings section. If you need to set a parameter which doesn't have a dedicated wizard option, you can use the Custom HTML Parameters pod in the Global Configuration and Playback Settings section.

Additional Advanced Information

By default, the slideshow is embedded into the `<div>` which preceeds it in the slideshow HTML code block. If you wish to embed a slideshow into some other `<div>` in your page, you can explicitly pass an `id` parameter in the initialization object. This should match the `id` attribute on the slideshow `<div>`:

```
<div id="slideshowHolder"></div>
...
```

```
XMLFlashSlideshow_v4({id:'slideshowHolder', ... });
```

The slideshow embed script uses the industry-standard SWFObject embed method (the same method used by Dreamweaver to embed SWFs). The included version is 2.2.

All of the passed parameters which aren't specifically used for some other purpose (e.g. `width` and `height`) are passed to the slideshow SWF as initialization parameters (via flashvars). The HTML parameter object can also be used to specify SWF embed parameters, such as `wmode`, `quality`, etc. Usually, the only parameter which would need to be modified would be wmode. To pass one of these embed parameters, you must prepend `_param_` to the parameter name; for example, to override the `wmode` setting you would use `_param_wmode` as the parameter name. For more information about the available param settings, see Step 1 in the SWFObject documentation.

Note: The slideshow by default always uses the transparent `wmode` setting. This allows any HTML content behind the slideshow to show through. The only downsides to the transparent `wmode` are that it can reduce rendering performance and can occasionally have bugs in certain configurations. Additionally, a `wmode` of `window` must be used if you wish to utilize color correction. To explicitly set the `wmode` to `window` or `opaque`, include a `_param_wmode` value in the configuration object. This can be done directly in the wizard by using the Custom HTML Parameters pod in the Global Configuration and Playback Settings section.

By default, the path to the slideshow SWF path is never passed in the slideshow HTML. Instead, the embed script detects the path to the supporting files folder based on the `slideshow.js` file's `<script>` reference and determines the appropriate SWF path. If for some reason this fails (as evidenced by right-clicking on the slideshow area and seeing 'Movie Not Loaded...'), you can manually specify the correct path through the `swf` parameter in the Slideshow HTML configuration object.

Note: The `showErrorBar` parameter (default = `true`) can be set to `false` to disable the 'Asset Loading Error' notification bar.

Developer Reference > **Overview of XML Structure**

Overview of XML Structure

XML configuration files have the following general format:

```
<?xml version="1.0" encoding="UTF-8"?>
<slideshow version="4" name="test0">
    <htmlTemplateMetaData ... />
    <styles>
        ...
    </styles>
    <effects>
        ...
    </effects>
    <kenburns>
        ...
    </kenburns>
    <ui>
        ...
    </ui>
    <galleries ... >
        ...
    </galleries>
</slideshow>
```

The first lines is the encoding declaration. This ensures that foreign characters can be used without issues; it should not be removed.

Using HTML in Various String Values

Most strings in the slideshow (such as gallery and image titles and descriptions) can contain basic HTML, including formatting tags and <img /> tags. However, this HTML must be encoded before being added to the XML. For instance, the following replacements must be made:

Entity	Proper Replacement

"	&quot;
<	&lt;
>	&gt;

Instead of manually replacing all of these values, you can instead use free HTML encoders available online. For example, see this encoder. It takes HTML content and encodes all of the necessary entities. If using a language such as PHP, you can do this with the `htmlspecialchars` function.

The Main <slideshow> Node

Next is the main `<slideshow> ... </slideshow>` node, which holds all of the slideshow configuration information. There are two attributes on this node:

Attribute	Description
`version`	Set to `4`; this is just used to clarify which version of the slideshow configuration file format the document is using. It has no direct effect on the slideshow.
`name`	This is the name associated with the slideshow. It is used to identify each slideshow when you are given the option to copy an existing slideshow when creating a new one.

The rest of the XML document is made up of several subnodes which are placed inside the `<slideshow>` holder node. An overview of each is given below.

The <htmlTemplateMetaData> Node

This section of the file is generated by the wizard. It is used when templating to include all Slideshow HTML initialization parameters in the XML configuration file, so that all templating data is centralized.

Details in The <htmlTemplateMetaData> section.

The <styles> Node

This section contains all of the global style selectors. These specify global settings (which have no direct visual representation) as well as default styles for various visual components.

Details in The <styles> section - Styles and Configuration.

The <effects> Node

This section contains the definitions of all reusable user-interface effects.

Details in The <effects> section - UI Effects.

The <kenburns> Node

This section contains the definitions of all reusable Ken Burns effects (which incorporate both panning and zooming and can be applied to slides).

Details in The <kenburns> section - Panning and Zooming Effects.

The <ui> Node

This section contains the Layout XML, which defines the layout and user interface.

Details in The <ui> section - User Interface.

The <galleries> Node

This section contains all of the actual image data.

Details in The <galleries> section - Image Data.

The <htmlTemplateMetaData> section

This section of the file is generated by the wizard. It holds XML representations of various HTML initialization parameters, such as: the background color, alpha, etc; the dimensions; and the preloader configuration. It is used when templating to include these Slideshow HTML initialization parameters in the XML configuration file, so that all templating data is centralized. All parameters are prepended with `p_` to clarify that they are meant to be used on the initialization object. Here is an example representation of the node.

```
<htmlTemplateMetaData note="THESE PARAMETER VALUES ARE FOR TEMPLATING ONLY; YOU MUST UPDATE THEM IN YOUR
HTML PAGE FOR THEM TO SHOW UP PROPERLY!" p_backgroundColor="#C5C5C5" p_preloaderTextColor="#000000"
p_height="400" p_width="600" p_backgroundAlpha="0.5"/>
```

Please note: These values are only for templating. Changing them does NOT affect the actual initialization parameters.

For more information on the Slideshow HTML parameters, see this topic.

Generally, this node is not manipulated directly; the wizard handles it.

The <styles> section - Styles and Configuration

This section contains all of the global style selectors. The term "style" is used rather loosely; these specify global settings (which have no direct visual representation) as well as default styles for various visual components. To learn more about the different types of global style selectors, see the Element Styling topic.

The section has a very simple format. Each separate group of settings has a "selector" name. For elements with a visual representation, this is the element name (e.g. `Button`). For general configuration settings, such as the image loading settings, the corresponding class name is used as the selector. In both cases, the selector is used to name a node which goes in the `<styles>` section. For example, the `com.dwuser.ss4.managers.ImageManager` class holds the image loading settings, and is represented in the styles section by an `<ImageManager ... />` node. Each property for each selector is added as an attribute on the corresponding selector's node.

It is best to illustrate with an example. Here is an example of a very simple `<styles>` section:

```
<styles>
    <Application themeColor="#3399FF"/>
    <IndividualSlideImage borderColor="#990000" borderThickness="1" backgroundColor="#FFFFFF"
useSpinner="false"/>
    <IndividualSlide useReflection="true"/>
    <AudioIntro errorReporting="true" stopOnGalleryOpen="false" autoplay="true"/>
</styles>
```

This example first includes a customized `Application themeColor` property (which is applied to the slideshow as a whole). Next, some special settings such as a border have been configured for the each slide's image using the `IndividualSlideImage` selector. In the third line, the refelection has been enabled for each slide using the `useReflection` property on the `IndividualSlide` selector. Finally, some parameters have been set for the intro-level audio.

For more information about all elements and supported non-visual selectors as well as their supported attributes/properties, see the API Reference. Again, for more about the different types of global style selectors, see the Element Styling topic.

Common Configuration Selectors

Here are some common style selectors which control the slideshow's configuration and their associated classes (see the API Reference):

Selector Name	Class	Description
ImageManager	com.dwuser.ss4.managers.ImageManager	Controls the way the images are loaded and managed, including such options as the loading mode.
IndividualSlideImage	com.dwuser.ss4.ui.IndividualSlideImage	Represents the image displayed on each slide; can be customized to add such features as dropshadow, matte, and border.
IndividualSlide	com.dwuser.ss4.ui.IndividualSlide	Represents each slide in the slideshow. Controls such options as the image scaling mode, the image alignment, and effects such as the reflection. In conjunction with WatermarkImage, can be used to enable a watermark.
WatermarkImage	com.dwuser.ss4.ui.WatermarkImage	Represents the watermark displayed on each image if `useWatermark` has been enabled on the `IndividualSlide` selector.
Thumb	com.dwuser.ss4.ui.Thumb	Represents the thumbs displayed on the Thumbnails element; the class reference includes more information about selectors for other thumb states (hover, selected, etc).
NumberBarTile	com.dwuser.ss4.ui.NumberBarTile	Represents the buttons/tiles displayed on the NumberBar element; the class reference includes more information about selectors for other button states (hover, selected, etc).
Application	mx.core.Application	Represents the base class used for the slideshow; common settings used on this selector include `themeColor` (establishes global default theme color) and font settings (`fontFamily`, `fontSize`, `color`, etc).
ToolTipAppearanceManager	com.dwuser.ss4.managers.ToolTipAppearanceManager	Controls the behavior of all tooltips used throughout the slideshow (e.g. whether they track the mouse, how long they remain visible, etc).
HTMLToolTip	com.dwuser.ss4.ui.extras.HTMLToolTip	Controls the apperance (colors, fonts, etc) of all tooltips used throughout the slideshow.

Developer Reference > **The <effects> section - UI Effects**

The <effects> section - UI Effects

This section contains the definitions of all reusable user-interface effects. To learn more about using user interface effects, see the Using UI Effects topic; you should review this topic before proceeding.

Because the same effects are often used more than once, UI effects are defined in one central location, the `<effects>` section. Each of the defined effects is assigned a unique ID identifier. When an effect is needed in the layout, an effect is referenced using its ID value in the following format:

```
<SomeElement ... hideEffect="{effect.effectIDValue}" ... />
```

In this example, a hide effect is specified.

To begin, here is an example of an `<effects>` section with two effects:

```
<effects>
     <effect id="simpleFadeIn" type="sequence">
          <Fade duration="4000" alphaFrom="0" alphaTo="1" easingFunction="Easing(Back.easeIn)"/>
     </effect>
     <effect id="blurBlinds" type="parallel">
          <Blur duration="4000" blurXTo="10" blurYFrom="0" blurXFrom="0" blurYTo="10"/>
          <Fade duration="4000" alphaTo="0" alphaFrom="1"/>
     </effect>
</effects>
```

The `<effects>` section is used by the slideshow to populate the effect library. All UI effects are managed by the `com.dwuser.ss4.managers.EffectLibrary` class. You should review this class (in the API Reference) for more information about how effects are managed.

The XML Structure

The `<effects>` section is comprised of one or more `<effect>` nodes. Each of these `<effect>` nodes defines a separate effect. Each `<effect>` has two required attributes:

Attribute	Description
`id`	Required; specifies the unique ID value with which the effect will be referenced elsewhere (see above).
`type`	Required; specifies if the subeffects which comprise the effect play in sequence or in parallel (all at the same time). Supported values are: ▪ `parallel` - Specifies that the subeffects which comprise the effect should play at the same time, in parallel. ▪ `sequence` - Specifies that the subeffects which comprise the effect should play in ordered sequence (according to the order in which they appear in the XML definition). *Note: Effect types are covered in the Using UI Effects topic referenced above.*

As covered in the Using UI Effects topic referenced earlier, each UI effect is comprised of one or more sub-effects. In the XML, each sub-effect is represented by a single node whose name is the effect name. Each of these sub-effect nodes is placed within the `<effect>` node. For example, the Blur effect is represented in the example above by a `<Blur ... />` node. The attributes on each sub-effect node set the properties of that sub-effect.

To learn about all of the available sub-effects, see the `com.dwuser.ss4.managers.EffectLibrary` class in the API Reference. This lists the available sub-effects as well as the class to reference for all of the available attributes.

The <kenburns> section - Panning and Zooming Effects

This section contains the definitions of all reusable Ken Burns (KB) effects (which incorporate both panning and zooming and can be applied to slides). To learn more about using KB effects, see the Using the Ken Burns Effect Editor topic; you should review this topic before proceeding.

Because the same KB effects are often used more than once in a given slideshow, KB effects are defined in one central location, the `<kenburns>` section. Each of the defined effects is assigned a unique ID identifier. When an effect is needed for a slide, a KB effect is referenced using its ID value in the following format:

`<gallery ... kbEffect="`**`effectIDValue`**`" ... >`
or
`<img ... kbEffect="" ... />`

You can also specify multiple effect ID's (separated by commas), and one will be chosen randomly each time. For more information about how to utilize the defined KB effects, see the <galleries> section documentation.

To begin, here is an example of an `<kenburns>` section with two effects:

```
<kenburns>
     <kbEffect id="NWtoSE" startX="-111" startY="-34" startZoom="1.2" endX="105" endY="57" endZoom="1"
delay="500" duration="5000" easing="Easing(Quadratic.easeInOut)"/>
     <kbEffect id="SEtoNW" startX="100" startY="40" startZoom="1.2" endX="-100" endY="-50" endZoom="1"
delay="500" duration="5000" easing="Easing(Quadratic.easeInOut)"/>
</kenburns>
```

The slideshow converts each `<kbEffect>` node in the `<kenburns>` section into a `com.dwuser.ss4.utils.KBEffect` instance. You should review this class (in the API Reference) for more information about the underlying architecture and information about all of the various settings.

The XML Structure

The `<kenburns>` section is comprised of zero or more `<kbEffect>` nodes (no effects are required to be defined). Each of these `<kbEffect>` nodes defines a separate KB effect. Each `<kbEffect>` has one required attribute -- it's ID value -- and several important but optional settings:

Attribute	Description
id	Required; specifies the unique ID value with which the effect will be referenced elsewhere (see above).
startX	Optional; specifies the x-offset from the center of the image which will be treated as the center when the panning/zooming effect starts. For example, a value of -100 makes the panning effect start 100 pixels to the left of the image's center. Default value of 0
startY	Optional; specifies the y-offset from the center of the image which will be treated as the center when the panning/zooming effect starts. For example, a value of -100 makes the panning effect start 100 pixels above the image's center. Default value of 0
startZoom	Optional; specifies the zoom factor (with 1 being the original size) of the image when the panning/zooming effect starts. Default value of 1.2
endX	Optional; specifies the x-offset from the center of the image which will be treated as the center when the panning/zooming effect completes. For example, a value of 100 makes the panning effect complete 100 pixels to the right of the image's center. Default value of 0
endY	Optional; specifies the y-offset from the center of the image which will be treated as the center when the panning/zooming effect completes. For example, a value of 100 makes the panning effect complete 100 pixels below the image's center. Default value of 0
endZoom	Optional; specifies the zoom factor (with 1 being the original size) of the image when the panning/zooming effect completes. Default value of 1.2
delay	Optional; specifies, in milliseconds, the delay before the KB effect begins to play. This can allow time for the slide transition to fully complete. Default value of 500

duration	Optional; specifies, in milliseconds, the duration of the KB effect. Default value of 5000 (5 seconds)
easing	Optiona; specifies the easing equation used to play the effect. To learn more about supported values, see the easing property on the KBEffect class referenced above. The default value is Easing(Quartic.easeInOut)

To disable panning for a given KB effect, set startX, startY, endX, and endY to 0. To disable zooming for a given KB effect, set startZoom and endZoom to 1. All of the settings mentioned here are described in more detail in the com.dwuser.ss4.utils.KBEffect class in the API Reference.

The <ui> section - User Interface

The <ui> section contains the Layout XML, which defines the layout and user interface. This topic only provides a general overview, as the detailed XML reference is in the API Reference (more information on this below).

First, to understand general slideshow user interface and layout principles, you should review the following topics in the Layout / Presentation section:

- Understanding and Managing Layout Views - Details what Layout Views are and how to use them.
- Layout Element Overview - Covers basic principles of using elements within a layout.
- Element Positioning and Sizing - Explains how to use the various positioning and sizing methods which are available.
- Element Styling - Explains the different ways styles can be applied to elements.
- Available Elements - Covers the many user interface elements which can be used.

Next, you should review in detail the `com.dwuser.ss4.ui.LayoutManager` class in the API Reference. This is the class which transforms the Layout XML into a fully-functioning user interface for the slideshow. The class reference has a great deal of information about the structure of the layout XML.

Once you have reviewed these topics, you should have a sound knowledge of how the <ui> section of the configuration file is constructed. Here is a review of the most important principles:

Important Principle Review

The `<ui>` holder node should contain one node - the `<views>` node. The `<views>` node holds the definitions for all of the layout views, each one implemented by a single `<view> ... </view>` node.

Each `<view>` node needs a `type` and `id` attribute; if `type` is omitted, it defaults to `normal`. The `id` can be omitted if the `type` has been explicitly defined to be `underlay` or `overlay`. If the `type` is `normal`, the `initial` attribute must be specified as well, with only one `<view>` using `initial="`**`true`**`"`. There are many other special attributes which can be used on the `<view>` nodes; these are all covered in the `LayoutManager` class.

Each `<view>` holds one or more child element nodes. The name of each node corresponds to the element name -- e.g. the ImagesStandard element is implemented by an `<ImagesStandard ... />` node. For nestable components (e.g. layout containers), child nodes are added inside the parent node. Properties and styles are set on the element via attributes of the corresponding name -- e.g. the label on a button (the `label` property) is set via `<Button`

`... label="My Label" ... />`. All available properties and styles are for each element are detailed in its implementing class in the API Reference; for the corresponding element classes, see the Available Elements reference. The order of definition of the element nodes in the XML file controls their relative stacking depth.

Here is an example of a slideshow UI's Layout XML, utilizing many of the mentioned features:

```
<ui>
    <views>

        <view type="underlay">
            <BorderBox left="0" right="0" top="0" bottom="0"
                backgroundColor="Bind{slideshow.currentPreSyncSlide.dominantColorLight}"
                borderThickness="0" borderStyle="solid"
                backgroundChangeEffectDuration="1700"
            />
        </view>

        <view type="normal" id="mainView">
            <!-- Note the relative positioning here; the image display takes up all but the bottom 100px, and the caption floats at the bottc
            <ImagesStandard left="0" top="0" right="0" bottom="100" />
            <Caption bottom="0" height="100" left="0" right="0" />
        </view>

        <view type="overlay">
            <!-- note the nested elements here: -->
            <VBox width="100%" height="100%" horizontalAlign="center" verticalAlign="middle">
                <SingleLoader />
                <GroupLoader />
            </VBox>
        </view>

    </views>
</ui>
```

Developer Reference > **The <galleries> section - Image Data**

The <galleries> section - Image Data

This section contains all of the actual image data. The <galleries> section has the following general format:

```
<galleries ... >
    <!-- One or more galleries in each slideshow (zero if using dynamic data source(s)) -->
    <gallery ... >
        <!-- One or more slides/images in each gallery -->
        <img ... />
    </gallery>
</galleries>
```

The main `<galleries>` node has various attributes which apply to all galleries. Inside this node are placed the `<gallery>` node(s). Each `<gallery>` node has various attributes which specify setttings specific to that gallery. Each `<gallery>` node is filled with one or more `<img ... />` nodes, each of which define a slide/image. If dynamic data is being used with a given gallery, then adding `<img ... />` nodes is optional. The following sections of this topic detail the format of each of these nodes.

The <galleries> node

The `<galleries>` node holds all of the gallery definitions. It has several properties / attributes which apply to all galleries:

Attribute / Property	Values	Description
`randomize`	`true` or `false`	Specifies if the order of the galleries in the slideshow (not their images) should be shuffled. Optional; defaults to `false`
`randomizeEveryGallery`	`true` or `false`	Specifies if the image order of all galleries should be randomized when they are opened. This allows for randomized playback. If enabled, this value overrides the gallery-specific `randomize` setting. Optional; defaults to `false`

autoplay	true or false	Specifies if autoplay should be initially enabled when the slideshow is opened. Optional; defaults to false
autoplayDelay	Autoplay Delay (see desc.)	Specifies the delay between images when autoplay is enabled. For details on the format used for this value, see the com.dwuser.ss4.managers.AutoplayManager class in the API Reference. This value can be overridden for individual galleries and images. Optional; has no default value. Required if autoplay is enabled.
autoplayLoop	true or false	Specifies if the slideshow should loop back to the first image in the first gallery when the final image in the last gallery has been reached. If the autoplayLoop setting is enabled for a single gallery, autoplay will loop within the gallery instead of skipping to the next gallery. Optional; defaults to false
audio	MP3 file URL	Specifies the URL of the MP3 background audio for the slideshow; its options are controlled via the AudioIntro selector (see the com.dwuser.ss4.pseudoStyleClasses.AudioIntro class in the API Reference). Optional; has no default value.
audioLabel	String label	If an audio attribute has been specified, this attribute specifies the audio label associated with the background audio. Optional; defaults to empty string.
effectType	Effect name(s); comma separated if more than 1.	Specifies the default transition effect between all slides if using the ImagesStandard element for image display. This value can be overridden for individual galleries. If using more than one effect, each of their names must be separated with a comma (no spaces). For more information about supported transitions, see Setting the Image Transition. For example: ...effectType="FadeStandard,Cube" ... Optional; defaults to FadeStandard
linkTarget	Link target (e.g.	Specifies the default link target for all image links. This value can be overridden for individual galleries and images.

	_self, _blank, etc).	Optional; defaults to _self

The <gallery> nodes

Each gallery is represented by a <gallery> node which is placed inside the <galleries> holder node. Each <gallery> node has the following properties / attributes:

Attribute / Property	Values	Description
id	Unique ID string	This **optional** value is only applicable if you plan to use dynamic data with the gallery or if you plan to use permalinks or javascript interfacing. When you create and use a dynamic data source, the data can be added to a new gallery or added to an existing gallery at runtime. If it is added to a new gallery, you won't have control over the gallery settings such as title, description, thumbnail, etc. If you want control over these settings, you must create a gallery in the wizard, assign it a linkage id value, then associate this gallery linkage ID with your dynamic data source in the External Image Data section. Then, when the dynamic data is loaded, the image information will be added to the gallery you created and any explicitly-specified parameters will not be modified. If an ID is specified, the gallery can later be referenced by using this value in a permalink (e.g. linking directly to a specific gallery) or a javascript interface call. See the linked topics for more information about how these features work. Note: This value must only contain letters, numbers, and underscores. Optional; has no default value. If omitted, the gallery can still be referenced in permalinks and javascript interface calls by using its zero-based numeric index.
fullPathPrefix	String	If specified, this optional value is prepended to all image URL's (the src attribute on each <img> node). This allows for redundant path structures to be listed only once, instead of bloating the configuration file by being listed for every image.
thumbPathPrefix	String	If specified, this optional value is prepended to all thumbnail URL's (the thumb attribute on each <img> node). This allows for redundant path structures to be listed only once, instead of bloating the configuration file by being listed for every image.
hqPathPrefix	String	*Generally should not be used.* If specified, this optional value is prepended to all high-quality image URL's (the hq attribute on each <img> node). This allows for redundant path structures to be listed only once, instead of bloating

		the configuration file by being listed for every image.
videoPathPrefix	String	*Generally should not be used.* If specified, this optional value is prepended to all slide video URL's (the video attribute on each <img> node). This allows for redundant path structures to be listed only once, instead of bloating the configuration file by being listed for every image.
title	String	This is the title value associated with the gallery. This value will be displayed in any gallery selection elements as the gallery's title. HTML can be included, but must be encoded (see Overview of XML Structure). Optional, but recommended. If omitted, has no default value.
description	String	This is the text description value associated with the gallery. This value will be displayed to provide information about the gallery in some gallery selection elements. HTML can be included, but must be encoded (see Overview of XML Structure). Optional, but recommended. If omitted, has no default value.
thumb	String	This is the URL to the gallery's thumbnail image. This image is displayed with the gallery in various gallery selection elements. Optional, but recommended. If omitted, has no default value.
linkTarget	Link target (e.g. _self, _blank, etc).	The link target represents the window in which image links (links applied to images in the gallery) will be opened. This is the default value for all images in the gallery. It can be overridden on a per-image basis by setting the link target on an individual image. Setting this value overrides the global default on the <galleries> node. The default value of _self will open the link in the current window. Using _blank will open the link in a new browser window. Using _parent will open the link in the parent page if using frames. Specifying any other value will control a specific frame in a frameset. **Tip:** For control over the appearance and size of the new window, you can use a javascript call in the image link to open a true popup window. In the wizard, you can use the Popup Link Generator. Optional; defaults to _self
defaultLink	Link URL	This is the default image link value for all images in the gallery. It is only applicable for images that have no per-image link defined; on those images, the image-specific value will be used as the image's link. For more information on the format of this attribute, see the link attribute for the <img> node below.

		Note that replacement values are very helpful in this attribute (see the `<img>` node's `link` attribute for more information about replacement values).
`defaultTitle`	String	This is the default image title value for all images in the gallery. It is only applicable for images that have no per-image title defined; on those images, the image-specific value will be used as the image's title. For more information on the format of this attribute, see the `title` attribute for the `<img>` node below. Note that replacement values are very helpful in this attribute (see the `<img>` node's `title` attribute for more information about replacement values).
`defaultCaption`	String	This is the default image caption/description value for all images in the gallery. It is only applicable for images that have no per-image caption defined; on those images, the image-specific value will be used as the image's caption. For more information on the format of this attribute, see the `caption` attribute for the `<img>` node below. Note that replacement values are very helpful in this attribute (see the `<img>` node's `caption` attribute for more information about replacement values).
`defaultToolTip`	String	This is the default image tooltip value for all images in the gallery. It is only applicable for images that have no per-image tooltip defined; on those images, the image-specific value will be used as the image's tooltip. For more information on the format of this attribute, see the `toolTip` attribute for the `<img>` node below. Note that replacement values are very helpful in this attribute (see the `<img>` node's `toolTip` attribute for more information about replacement values).
`password`	String	Optionally specifies a gallery password. If a gallery password has been specified, the user will receive a password prompt when attempting to open the gallery, and will be required to enter the password correctly. **Note:** In the wizard, this value is encrypted using a proprietary encryption algorithm. If you generate your own XML, you must use un-encrypted string value. Optional; has no default value.
`randomize`	`true` or `false`	Specifies if the gallery should be randomized each time it is opened. This allows for randomized playback. Overrides the global setting on the `<galleries>` node. Optional; defaults to `false`

`autoplayDelay`	Autoplay Delay (see desc.)	Specifies the delay between images when autoplay is enabled. For details on the format used for this value, see the `com.dwuser.ss4.managers.AutoplayManager` class in the API Reference. This value overrides the global default on the `<galleries>` node and can be overridden for individual images. Optional; has no default value. Required if Autoplay is being used.
`autoplayLoop`	`true` or `false`	Enables/disables looping within the gallery. If enabled, autoplay will jump back to the first image in the gallery when the final image is reached. If disabled, autoplay will jump to the first image in the next gallery when the final image is reached. Optional; defaults to `false`
`effectType`	Effect name(s); comma separated if more than 1.	Specifies the default transition effect between all slides if using the ImagesStandard element for image display. This value overrides the global default on the `<galleries>` node and can be overridden for individual images. If using more than one effect, each of their names must be separated with a comma (no spaces). For more information about supported transitions, see Setting the Image Transition. For example: `...effectType="FadeStandard,Cube" ...` Optional; defaults to `FadeStandard`
`audio`	MP3 file URL	Specifies the URL of the MP3 gallery-level audio. This is the audio which plays only while this gallery is open. The audio file must be in MP3 format; for necessary formatting guidelines, see Avoiding "Chipmunks" Audio: MP3 Formatting Guidelines. The URL must be relative to the HTML page which holds the slideshow. **Note:** If you are loading audio from a **different domain** than the one on which the slideshow is displayed, you will be affected by Flash Player crossdomain loading security restrictions. To learn more, see this topic. **Note:** If you're audio plays back at a **distorted speed**, like "**chipmunks**", you need to review the MP3 formatting guidelines referenced above. The options for gallery-level audio are controlled via the `AudioGallery` selector (see the `com.dwuser.ss4.pseudoStyleClasses.AudioGallery` class in the API Reference). Optional; has no default value.
`audioLabel`	String label	If an `audio` attribute has been specified, this attribute specifies the audio label associated with the gallery-level audio.

		Optional; defaults to empty string.
kbEffect	KB Effect ID(s); comma separated if more than 1. Also, Random can be used.	Specifies a default Ken Burns effect (or effects) for all images in the gallery. Ken Burns effects (KB effects) are packages which incorporate both panning and zooming. You can learn more about KB effects in the <kenburns> section reference. If you specify an effect for this gallery-wide default, it will be used by each image in the gallery unless you override it on an individual image node via its kbEffect attribute. You can specify multiple KB effect ID values; when doing this, separate each ID with a comma (no spaces). One of these multiple effects will be randomly chosen each time an effect is to be applied. If you specify Random as the attribute value, an effect will randomly be chosen from among **all** defined effects. Specifying Cycle will cause all defined effects to be cycled through as effects are needed. Optional; has no default value.

The <img> nodes

Each slide / image in a gallery is represented by an <img> node which is placed inside the <gallery> holder node. Each <img> node has the following properties / attributes:

Note: See the Image Property Editing topic for more information and/or illustrations for each property.

Attribute / Property	Values	Description
title	String	This is the title associated with the image. Along with the description/caption, these are generally the two most commonly used (and important) properties you'll set for each image. This is the title value which will be displayed in the slideshow's Caption element, or in any dynamic text element for {image.title} . HTML can be used, but must be encoded (see Overview of XML Structure). Shortcut Text (AKA Replacment Values) can be used in this property; see this topic for more information about replacement values. Optional, but recommended; if omitted, defaults to an empty string.
caption	String	This is the description (or caption) associated with the images. Along with the title, these are generally the two

		most commonly used (and important) properties you'll set for each image. This is the description/caption value which will be displayed in the slideshow's Caption element, or in any dynamic text element for {image.caption} . HTML can be included, but must be encoded (see Overview of XML Structure). Shortcut Text (AKA Replacment Values) can be used in this property; see this topic for more information about replacement values. Optional; defaults to empty string.
toolTip	String	This is the tooltip text which will be displayed when a user hovers over the image. HTML can be included, but must be encoded (see Overview of XML Structure). Shortcut Text (AKA Replacment Values) can be used in this property; see this topic for more information about replacement values. Optional; no tooltip is displayed if omitted.
id	Unique ID String	This **optional** setting is only for advanced users. It specifies a unique ID by which can be used to reference the image when creating permalinks or making javascript interface calls. As described in permalinks topic, specifying an ID makes it simple to link directly to a specific image in a slideshow. Note: Only letters, numbers and underscores may be used in this value. Optional; if omitted, the image can still be referenced in permalinks and javascript interface calls using its numeric zero-base index.
src	Image URL	This is the URL of the slide's image. This can be any valid image URL, pointing to a JPEG, GIF, PNG, or even an SWF animation. Keep in mind that whenever you load from a remote site you may run into Flash Player Cross-domain Loading Restrictions. The URL must be relative to the HTML page holding the slideshow. If a fullPathPrefix attribute was specified on the parent <gallery> node (see above), it will be prepended to the URL unless one of the following is true: ■ The URL is absolute (e.g. starts with http://, https://, or file://) ■ The URL has {noprefix} at the start of it. This tells the slideshow not to apply any prefix to it. Required.

`thumb`	Thumbnail URL	This can be any valid image URL, pointing to a JPEG, GIF, PNG, or even an SWF animation. Keep in mind that whenever you load from a remote site you may run into Flash Player Cross-domain Loading Restrictions. The URL must be relative to the HTML page holding the slideshow. Optional, but highly recommended. If you omit this parameter, the full-size image will be loaded, scaled down, and used as the thumbnail. However, it is **highly recommended** that you create separate thumbnail images because using the full-size images often results in lengthly loading times, delayed previews, and pixelation due to the high level of scaling. Using separate thumbnail images ensures an all-around better user experience. You can generate thumbnails automatically for the entire gallery by using the Thumbnail Generator in the wizard. If a `thumbPathPrefix` attribute was specified on the parent `<gallery>` node (see above), it will be prepended to the URL unless one of the following is true: ▪ The URL is absolute (e.g. starts with `http://`, `https://`, or `file://`) ▪ The URL has `{noprefix}` at the start of it. This tells the slideshow not to apply any prefix to it.
`hq`	Image URL	*(Advanced parameter for experienced users only)* (Optional; usually unnecessary) This only applies to certain layouts in the professional version which contain one of the following image display elements: ImagesCoverFlow, ImagesScatter, ImagesScrollList, or ImagesStarfield. Its applicability is also dependent on the image preloading mode used (especially the 3D preloading mode). Generally, it is not the best option to use. However, in some cases, this URL represents the full-size high-quality (HQ) version of the image. This is the 'upgraded' version of the image which is loaded only when the image becomes the currently selected one. In these cases, you will set the main Image URL field to a full-size but lower quality (hence smaller file size and quicker loading) version of the image. Or, you can use a thumbnail version as the full size image but specify explicit dimensions for the HQ image below. These dimensions will be used to scale the thumb up as if it were the full-size image, displaying it until the HQ image loads. If a `hqPathPrefix` attribute was specified on the parent `<gallery>` node (see above), it will be prepended to the URL unless one of the following is true: ▪ The URL is absolute (e.g. starts with `http://`, `https://`, or `file://`) ▪ The URL has `{noprefix}` at the start of it. This tells the slideshow not to apply any prefix to it.

w	Numeric value	*(Advanced parameter for experienced users only)* If using the HQ Image URL as described above, in some cases you must specify the explicit dimensions of the full-size image so the slideshow knows before the image actually loads (see HQ Image description for more info). This specifies the width in pixels.
h	Numeric value	*(Advanced parameter for experienced users only)* If using the HQ Image URL as described above, in some cases you must specify the explicit dimensions of the full-size image so the slideshow knows before the image actually loads (see HQ Image description for more info). This specifies the height in pixels.
link	Link URL	Specifies the link which is opened when the user clicks on the image. Specifying a value here will override any gallery-wide defaults (the <gallery> defaultLink property). The URL can be either a relative path (relative to the slideshow's HTML page), or a full http:// URL. You can also call javascript by using links in the following format: javascript:void(myJavascriptFunction('param1', 'param2')); If you want to have the link open in a true popup window (which gives you control over the size and appearance of the window), you must use a special javascript link. The Popup Link Builder in the wizard will automatically generate this type of link for you. See the Using the Popup Link Builder section for more information. **Important note:** Whenever using a popup link or a javascript link, you should set the linkTarget property to _self . Shortcut Text (AKA Replacment Values) can be used in this property; see this topic for more information about replacement values.
linkTarget	Link target (e.g. _self, _blank, etc).	Specifies the window in which the image link will be opened. Specifying a value here will override any gallery-wide defaults (the <gallery> linkTarget property). The default value of _self will open the link in the current window. Using _blank will open the link in a new browser window. Using _parent will open the link in the parent page if using frames. Specifying any other value will control a specific frame in a frameset. For control over the appearance and size of a new window, you must use a true popup window. See the link property directly above for more information about this. **Important note:** Whenever using a popup link or a javascript link, you should set the linkTarget to _self. Shortcut Text (AKA Replacment Values) can be used in this property; see this topic for more information about

		replacement values. Optional; defaults to _self
autoplayDelay	Autoplay Delay (see desc.)	When autoplay is enabled, this value specifies the delay between each image/slide. Because this is a slide-specific value, it configures how long the current image/slide is shown. You can also specify a value which stops autoplay immediately, or automatically jumps to the image link associated with this image. These values are specially constructed strings; for details on the format used for this value, see the com.dwuser.ss4.managers.AutoplayManager class in the API Reference. Optional; has no default value. Required if Autoplay is being used.
effectType	Effect name(s); comma separated if more than 1.	Specifies the default transition effect between all slides if using the ImagesStandard element for image display. This value overrides the global default on the <galleries> node and the gallery default on the <gallery> node. If using more than one effect, each of their names must be separated with a comma (no spaces). For more information about supported transitions, see Setting the Image Transition. For example: ...effectType="FadeStandard,Cube" ... Optional; defaults to FadeStandard
audio	MP3 file URL	Specifies the URL of the MP3 image-level audio. This is the audio which plays only while this image is open. The audio file must be in MP3 format; for necessary formatting guidelines, see Avoiding "Chipmunks" Audio: MP3 Formatting Guidelines. The URL must be relative to the HTML page which holds the slideshow. **Note:** If you are loading audio from a **different domain** than the one on which the slideshow is displayed, you will be affected by Flash Player crossdomain loading security restrictions. To learn more, see this topic. **Note:** If you're audio plays back at a **distorted speed**, like "**chipmunks**", you need to review the MP3 formatting guidelines referenced above. The options for gallery-level audio are controlled via the AudioImage selector (see the com.dwuser.ss4.pseudoStyleClasses.AudioImage class in the API Reference). Optional; has no default value.
audioLabel	String label	If an audio attribute has been specified, this attribute specifies the audio label associated with the image-level

		audio. Optional; defaults to empty string.
audioAutoplay	Omit attribute or true or false	If an audio attribute has been specified, this attribute can be used to override the global image-level audio configuration setting for autoplay. For example, you might have image-level audio on all slides and have autoplay disabled (the user has to click a play button when interested). However, on one slide, you want the audio to start playing automatically. To do this, you would add <img ... **audioAutoplay="true"** /> for that one slide. Optional; omitting the parameter prevents any manipuation. **Note:** This parameter is a **code-only parameter** and has no visual representation in the wizard; it must be added manually.
galleryAudioPlay	Omit attribute or true or false	This attribute can be used to force the gallery-level audio (if there is any) to play or pause when this image is opened. Not specifying an attribute (the default) will prevent any manipulation of the gallery-level audio. Specifying true will force the audio to play; specifying false will force the audio to stop/pause. Optional; omitting the parameter prevents any manipulation.
galleryAudioPosition	Numeric value	This attribute can be used to force the gallery-level audio (if there is any) to jump to a specific location when this image is opened. This can be used to sync gallery-level audio to a specific location for each slide. Not specifying an attribute (the default) will prevent any manipulation of the gallery-level audio; otherwise, a numeric value must be passed (representing milliseconds). Optional; omitting the parameter prevents any manipulation.
introAudioPlay	Omit attribute or true or false	This attribute can be used to force the intro-level audio (if there is any) to play or pause when this image is opened. Not specifying an attribute (the default) will prevent any manipulation of the intro-level audio. Specifying true will force the audio to play; specifying false will force the audio to stop/pause. Optional; omitting the parameter prevents any manipulation.
introAudioPosition	Numeric value	This attribute can be used to force the intro-level audio (if there is any) to jump to a specific location when this image is opened. This can be used to sync intro-level audio to a specific location for each slide. Not specifying an attribute (the default) will prevent any manipulation of the intro-level audio; otherwise, a numeric value must be passed (representing milliseconds).

		Optional; omitting the parameter prevents any manipulation.
`video`	Video URL; see desc.	This attribute can be used to add a video file to the slide. The video URL can point to any of the following types of files: - A standard FLV Flash Video File located within your site or hosted at some remote location (see cross-domain loading note below) - An H.264 MPEG4 high-definition video file located within your site - An RTMP streaming video URL (usually delivered through the Flash Media Server). **Very Important:** If using a relative URL, the URL must be relative to the supporting files folder, NOT the HTML page holding the slideshow. All other relative URLs in the slideshow (for images, audio, etc) are relative to the HTML page, whereas relative video URLs must be relative to the supporting files folder. If a `videoPathPrefix` attribute was specified on the parent `<gallery>` node (see above), it will be prepended to the URL unless one of the following is true: - The URL is absolute (e.g. starts with `http://`, `https://`, or `file://`) - The URL has `{noprefix}` at the start of it. This tells the slideshow not to apply any prefix to it. **Note:** If you are loading audio from a **different domain** than the one on which the slideshow is displayed, you will be affected by Flash Player crossdomain loading security restrictions. To learn more, see this topic. **Note for advanced users:** To learn about how to customize the appearance of the video player, see the `com.dwuser.ss4.video.FlexVideoPlayer` class in the API Reference. Optional; has no default value.
`videoAutoStart`	`true` or `false`	If a `video` URL has been set, specifies whether the video should automatically start when the slide / image opens. If left at the default value (`false`), the user must click on the slide to start the video. In these cases, you may want to specify a `toolTip` such as "Click to start video...". Optional; defaults to `false`
`kbEffect`	KB Effect	Specifies the Ken Burns effect (or effects) for this image. Ken Burns effects (KB effects) are packages which

	ID(s); comma separated if more than 1. Also, Random can be used.	incorporate both panning and zooming. You can learn more about KB effects in the <kenburns> section reference. Setting this attribute will override any default value set on the <gallery> node. If a gallery-wide default effect exists and you want to override that default on an image with 'no effect' (disabling the KB effect for that image), then use kbEffect="" or kbEffect="None" . Omitting the attribute on the <img> node causes the gallery-wide default to be inherited. You can specify multiple KB effect ID values; when doing this, separate each ID with a comma (no spaces). One of these multiple effects will be randomly chosen each time an effect is to be applied. If you specify Random as the attribute value, an effect will randomly be chosen from among **all** defined effects each time the slide / image is displayed. Specifying Cycle will cause all defined effects to be cycled through as effects are needed. Optional; has no default value, which causes the gallery-wide default to be inherited.

The <img> Node's Subnodes

There are also two types of nodes which can optionally be placed inside an <img> node. These nodes, the <note ... /> node and <text ... /> node, define overlay notes and overlay text respectively. Review each of the linked topics for more information about these two items.

Here is an example of an <img> node with two overlay notes and one overlay text. The text appears after 3 seconds and disappears 5 seconds later:

```
<img src="somefile.jpg" ... >
     <note x="50" y="50" width="100" height="60" toolTip="Some information about this spot on the
picture" />
     <note x="250" y="350" width="30" height="70" toolTip="Some information about this second spot on the
picture" />
     <text x="250" y="50" htmlText="A Special Text Label" delay="3000" visibleTime="5000" />
</img>
```

The next two sections provide more information about each of these two nodes.

Overlay Note Box (<note ... />) Information

To change how long the note boxes remain visible before automatically hiding, set the noteHideDelay property on the IndividualSlide style selector (see com.dwuser.ss4.ui.IndividualSlide in the API Reference). If you use a very large value for this setting, the boxes will remain visible after being activated instead of auto-hiding.

To customize the appearance of the boxes and their tooltips, you will use the NoteBox and NoteBoxToolTip style selectors respectively. For more information, see the com.dwuser.ss4.ui.extras.NoteBox and com.dwuser.ss4.ui.extras.NoteBoxToolTip classes.

To customize the behavior of the tooltips (e.g. the appearance delay and whether they track the mouse), you will use the ToolTipAppearanceManager style selector. For more information, see the com.dwuser.ss4.managers.ToolTipAppearanceManager class.

All overlay note instances are instances of the com.dwuser.ss4.ui.extras.NoteBox component. As such, the <note ... /> nodes in the <img> nodes which define the overlay notes are actually NoteBox instances. They support most of the attributes available for the NoteBox component; see the class reference for available parameters. The following attributes are especially important and most commonly used:

Attribute / Property	Values	Description
toolTip	String	Specifies the tooltip text which is displayed when the user hovers over the overlay note box.
x	Numeric value (in pixels)	Specifies the horizontal position of the box, in pixels.
y	Numeric value (in pixels)	Specifies the vertical position of the box, in pixels.
width	Numeric value (in pixels)	Specifies the width of the box, in pixels.
height	Numeric value (in pixels)	Specifies the height of the box, in pixels.

Overlay Text (<text ... />) Information

All overlay text instances are instances of the mx.controls.Label component. As such, the <text ... /> nodes in the <img> nodes which define the overlay text-boxes are actually Label instances. They support most of the attributes available for the Label component; see the class reference in the API Reference for available parameters.

The inherited showEffect and hideEffect attributes can be used if effects are desired if/when the text appears or disappears. For information about defining effects and the attribute format used to reference them, see the reference for the <effects> section.

The following attributes are especially important and most commonly used (note that the delay and visibleTime properties are special additions; they are NOT inherited from the Label class):

Attribute /	Values	Description

Property		
delay	Numeric value (in milliseconds)	Optionally specifies, in milliseconds, how long after the slide is created before the overlay text box will appear. If omitted, the text box will be visible as soon as the slide is created.
visibleTime	Numeric value (in milliseconds)	Specifies, in milliseconds, how long the text will be visible before it automatically hides. If omitted, the text won't ever automatically hide.
htmlText	String	Specifies the text displayed in the overlay text box. HTML can be included, but must be encoded (see Overview of XML Structure).
x	Numeric value (in pixels)	Specifies the horizontal position of the text box, in pixels.
y	Numeric value (in pixels)	Specifies the vertical position of the text box, in pixels.
color	Hexadecimal color (e.g. #FF0000)	Specifies the color of the text. Optional; default is #0B333C
fontFamily	Font name	Specifies the font used for the text. Optional; default is Verdana
fontSize	Numeric value (in pixels)	Specifies the size of the text, in pixels. Optional; default is 11

External Data Source Format

The format used for dynamic data source is quite similar to that used in the `<galleries>` node in the configuration file. To learn more about the supported formats, see Supported Dynamic Data Formats in the Using Dynamic Data section of the manual. Note that the `<galleries>` wrapper node used in the Multiple Galleries Format does **not** support any of the attributes of the main `<galleries>` node in the configuration file; these settings must be set in the main XML configuration file. However, each `<gallery>` node **does** support all of the `<gallery>` attributes detailed in the <galleries> section reference.

API Reference / Detailed UI Element Reference

Open in Separate Window *(Recommended!)*

Special Undocumented Style Properties

Due to the way the API documentation is generated, there are several special style properties available on elements with borders which are not listed in the API documentation. This topic covers these properties.

borderAlpha

For elements with borders, the `borderAlpha` style property can be used to specify the opacity of the border. Valid values range from 0.0 (fully transparent) to 1.0 (fully opaque). The default value is 1.0 (fully opaque).

highlightStrength

For elements with a background color, the `highlightStrength` style property can be used to add a highlight bevel to the background color. This colored overlay fades from the top to the bottom, starting at the opacity specified by `highlightStrength` and finishing at 0 at the bottom. Valid values range from 0.0 (fully transparent at the top; no highlight visible) to 1.0 (fully opaque highlight at the top). The default value of 0.0 equates to no highlight being used. The color of the highlight defaults to white (`#FFFFFF`), but can be customized through the `highlightColor` property.

Here is an example of an element with a highlight on its background (`highlightStrength` is 0.4 in this example):

highlightColor

If a highlight has been enabled through the `highlightStrength` style property, the `highlightColor` property specifies the color of that highlight. The default value is `#FFFFFF` (white).

Advanced Topics / Advanced Examples

This section of the manual covers various advanced topics which don't belong under any specific topic. Advanced examples are also included. Most of the topics are targeted at advanced users.

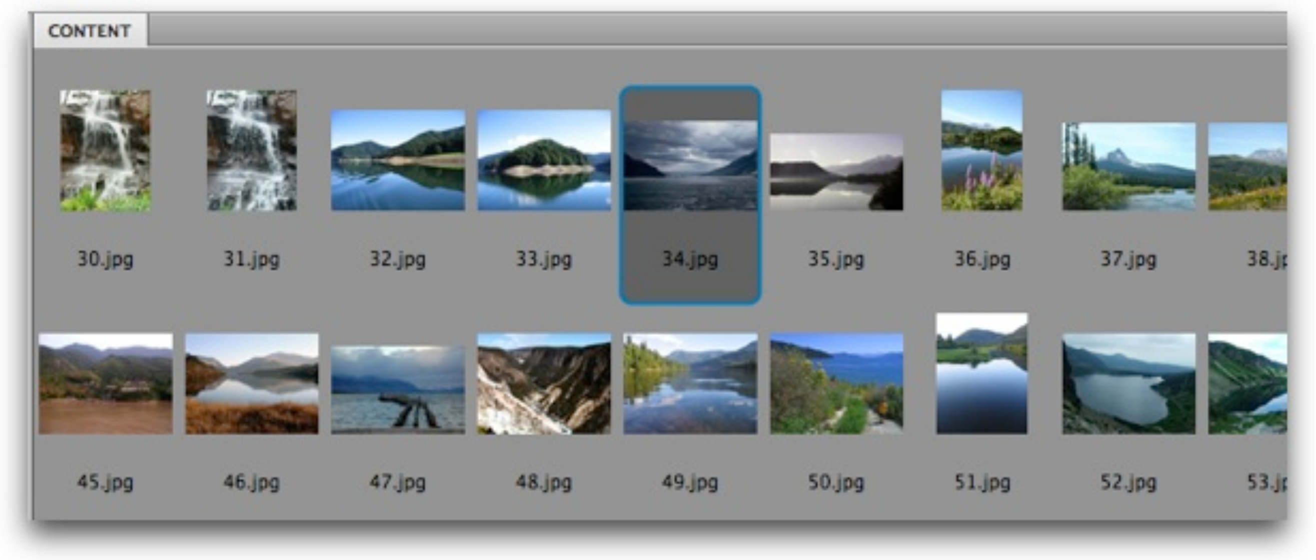
Image Metadata Integration

Image metadata is special data or information that is encoded into an image file. The information is stored as a part of the image file itself, rather than being stored externally and linked to that image file. Especially for users of Adobe Bridge, the ability to encode photo information directly into its file may be a convenient option.

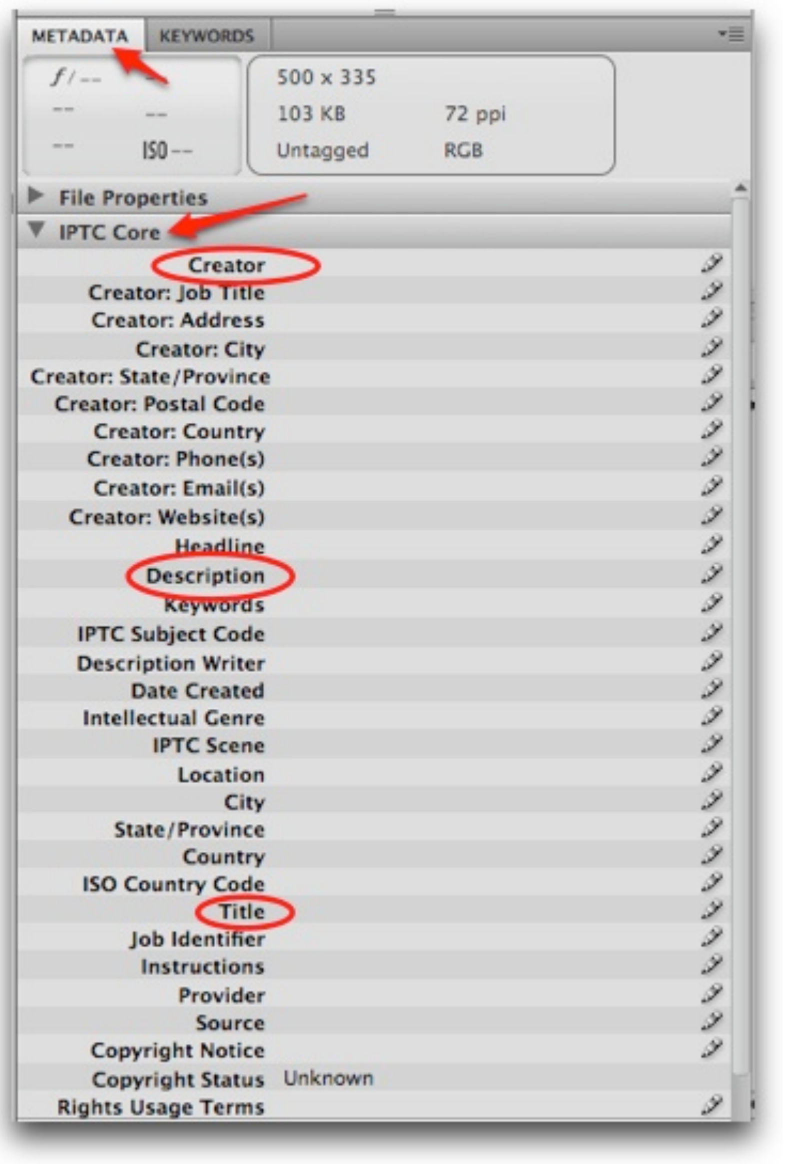
The XML Flash Slideshow v4 includes the option to read this metadata (in XMP [Extensible Metadata Platform] format) from images at runtime. Currently supported parameters are the Creator field, the Title field, and the Description field. Harnessing the power of metadata can allow clients to more easily add and remove images with titles and captions in their slideshows, or simply prevent unnecessary duplication of data entry when configuring image titles and captions.

Setting Metadata Parameters in Bridge

For this example, Adobe Bridge will be used to set metadata parameters. First, browse to and select the desired photo(s) in the Content panel:



Switch to the Metadata panel (Window > Metadata Panel) and expand the IPTC Core section:



f / --
-- --
-- ISO --

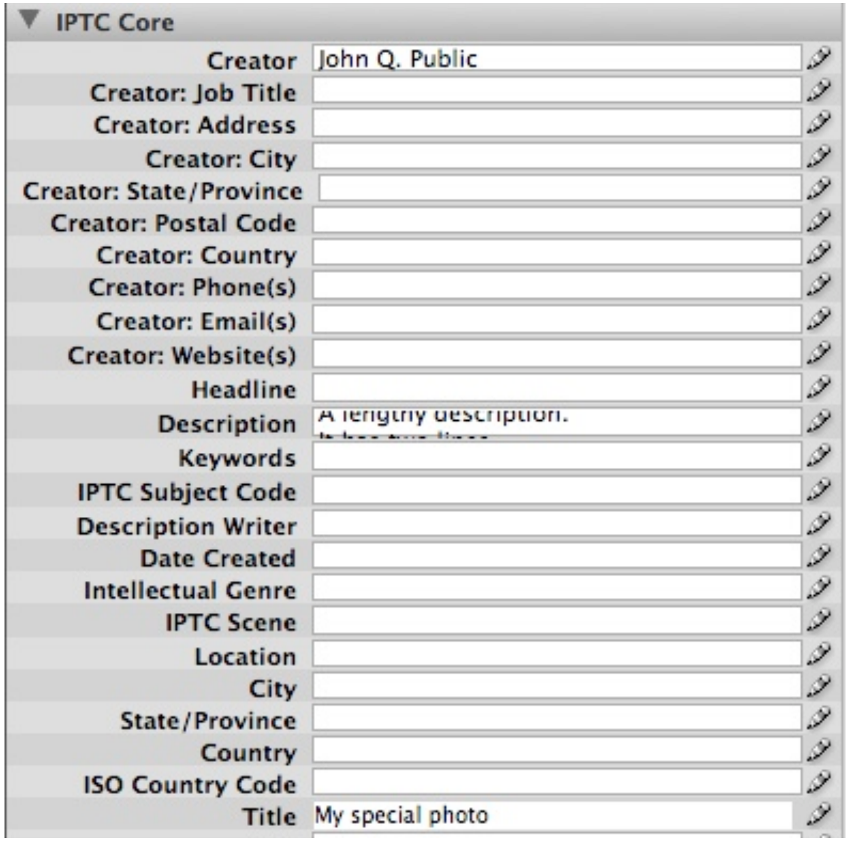
500 x 335
103 KB 72 ppi
Untagged RGB

▶ File Properties

▼ IPTC Core

Creator
Creator: Job Title
Creator: Address
Creator: City
Creator: State/Province
Creator: Postal Code
Creator: Country
Creator: Phone(s)
Creator: Email(s)
Creator: Website(s)
Headline
Description
Keywords
IPTC Subject Code
Description Writer
Date Created
Intellectual Genre
IPTC Scene
Location
City
State/Province
Country
ISO Country Code
Title
Job Identifier
Instructions
Provider
Source
Copyright Notice
Copyright Status Unknown
Rights Usage Terms

The three currently-supported properties have been highlighted -- Creator, Title, and Description. To edit a field, click in the area to the right of the property name and enter text:



▼ IPTC Core

Creator	John Q. Public
Creator: Job Title	
Creator: Address	
Creator: City	
Creator: State/Province	
Creator: Postal Code	
Creator: Country	
Creator: Phone(s)	
Creator: Email(s)	
Creator: Website(s)	
Headline	
Description	A lengthy description.
Keywords	
IPTC Subject Code	
Description Writer	
Date Created	
Intellectual Genre	
IPTC Scene	
Location	
City	
State/Province	
Country	
ISO Country Code	
Title	My special photo

Enter any relevant fields, then select a different photo in the Content panel. When you browse away from the photo whose metadata was just modified, you may be asked if you wish to save the metadata changes -- be sure to allow this.

Using the Metadata Within a Slideshow

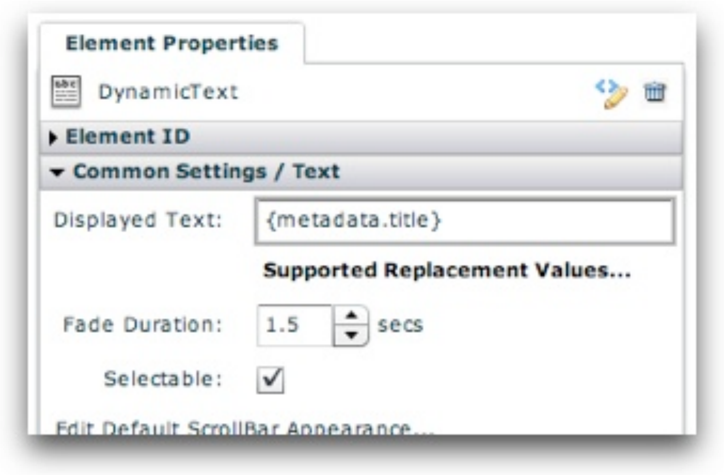
Back within your slideshow, metadata can be used just for specific photos, or used on dynamic text elements to display the available metadata for *all* photos (including ones from external data sources). In either case, one or more special replacement values are used to designate the inclusion of metadata:

- `{metadata.creator}` - This value will be replaced with the Creator metadata property
- `{metadata.title}` - This value will be replaced with the Title metadata property
- `{metadata.description}` - This value will be replaced with the Description metadata property
- **Tip:** On all of the above properties, a default value (in case the requested value is not yet available or doesn't exist) may be specified in the following

format: {metadata.description|**Default Value**} . After the property name, a pipe character is followed by the default value.

Generally, the easiest way to implement metadata into a gallery is to use the Default Image Title and Default Image Caption gallery properties. For example, setting a Default Image Title of {metadata.title|} would make the image title in the Caption element (and anywhere else {image.title} is used) display the metadata title if one exists, or an empty string if none exists.

For specific photos, you can use the replacement values in the Caption or Title properties for that given image (in the Image Properties dialog in the slideshow wizard). For more general inclusion, these replacement values are used in the DynamicText, DynamicScrollableText, or DynamicLinkButton elements. For example, the following DynamicText element is configured to display the available metadata-specified title:


Element Properties
DynamicText
Element ID
Common Settings / Text
Supported Replacement Values...
Edit Default ScrollBar Appearance...

Displayed Text: {metadata.title}

Fade Duration: 1.5 secs

Selectable: ☑

Custom Skin Files / Custom Buttons

This section of the manual details the process of creating custom buttons (AKA "skinning" buttons) and, for more advanced users, creating custom skin files. Basic button skinning is quick and simple and only involves creating an image file; it doesn't require any specialized software like Adobe Flash.

- Learn about Easy Image-Based Button Skinning

Creating skin file packages allows for a slightly higher level of customization. Custom fonts can be embedded, button skins can include transitions between states, and other asset classes such as icons and custom skin classes can be made available. Skin file packages are special centralized SWF packages that hold the embedded assets. Currently, Adobe Flash CS3, CS4, CS5 or higher must be used to create these skin files; the process is detailed in a subtopic:

- Using Adobe Flash CS3 or Flash CS4 or Flash CS5 (or higher)

Easy Image-Based Button Skinning

Easy image-based button skinning allows you to create and apply custom button images to your slideshow quickly and easily. First, it's important to understand the concept of buttons modes and states.

Button Modes and States

Most buttons in a slideshow only have **one mode** - they always show the same "icon" and always perform the same action when clicked. An example of a button with only one mode is the NextButton element. The NextButton always shows the same icon (usually a forward arrow), and always performs the same action (progressing the slideshow to the next image).

Other buttons, however, have **two modes**. These buttons can show two different "icons" and will perform two different actions depending on which mode they're in. An example of this type of button is the PlayPauseButton element. In the first of its modes, the PlayPauseButton shows a "play" icon and starts the slideshow playback when clicked. In the second of its two modes, it shows a "pause" icon and stops slideshow playback when clicked.

For buttons with two modes, the second mode is referred to as its **selected** mode. Thus, if a button is "selected", this means that it is currently in its second mode. (Note - If a button can be *toggled*, this is another way of saying that it has two modes.)

For each *mode* a button has, it has **four** corresponding ***states***. These states are:

State Name	Info
up	Represents the button when there is no user interaction. This its default/normal state.
over	Represents the button when the user hovers over it. (E.g. when they roll over a button.)
down	Represents the button when the user presses down the mouse on it. (E.g. when they click the button.)
disabled	Represents the button when it is disabled. (E.g. the PreviousButton is disabled when the first image is being displayed.)

Thus, buttons with **one mode** have **four states**, and buttons with **two modes** have **eight states**. Buttons with one mode have the following states: *up, over, down, disabled*. Buttons with two modes have the following states: *up, over, down, disabled, selectedUp, selectedOver, selectedDown, selectedDisabled*. The first four of the eight represent the skins for the button's first mode; the last four represent the skins for the selected (second) mode of the button.

The following table summarizes the states:

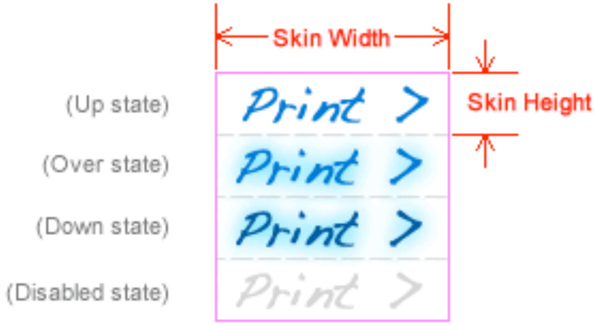
State	Info
	States which apply to all buttons/elements:
up	Represents the skin when there is no user interaction. This is its default/normal state.
over	Represents the skin when the user hovers over it. (E.g. when they roll over a button.)
down	Represents the skin when the user presses down the mouse on it. (E.g. when they click the button.)
disabled	Represents the skin when the element (button) is disabled. (E.g. the PreviousButton is disabled when the first image is being displayed.)
	States which only apply to toggled (selectable) buttons/elements:
selectedUp	*Only applies to the elements listed in the table above which utilize the selected state.* Represents the up state (as described above) when the button is selected.
selectedOver	*Only applies to the elements listed in the table above which utilize the selected state.* Represents the over state (as described above) when the button is selected.
selectedDown	*Only applies to the elements listed in the table above which utilize the selected state.* Represents the down state (as described above) when the button is selected.
selectedDisabled	*Only applies to the elements listed in the table above which utilize the selected state.* Represents the disabled state (as described above) when the button is selected.

Creation of Sprite Images

Simple button skinning uses an image technique called **sprites** to centralize all images for a skin in a single image file. Put simply, a sprite is one larger image that includes several smaller images positioned next to each other; each of the smaller images represents a single state for the skin. At runtime, the larger image is loaded and automatically split apart into its constituent states, simulating the use of separate image files for each state. Sprites make designing a skin much easier since all states are next to each other in a single design file. The technique also reduces bandwidth and server load at runtime because only one file must be transferred; this provides a faster and better user experience.

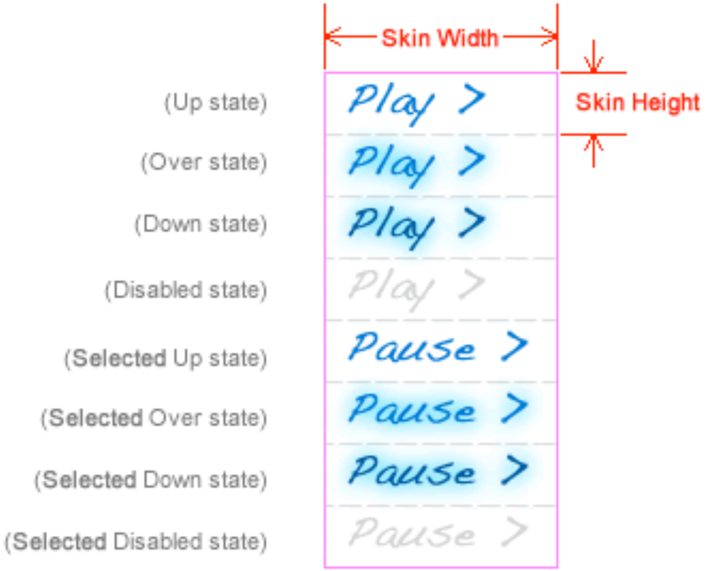
When creating an image-skinned custom button, the first step is to create one of these sprite images.

For slideshow button skins, the images for each state are stacked vertically in the sprite image. For one-mode buttons, the four state images are stacked like this:



The pink border represents the outer border of the sprite image (the image that is used as the skin image). The dashed gray lines represent the imaginary edges of each of the invidual state images. **Important! Note that all state images must be the exact same size.** This size in the example image is denoted with the Skin Width and Skin Height markers. These dimensions are important, as they determine where the image will be sliced apart when the skin is created; they must be entered in the wizard when creating an image skin (more information below).

For two-mode buttons, the eight state images are stacked like this:

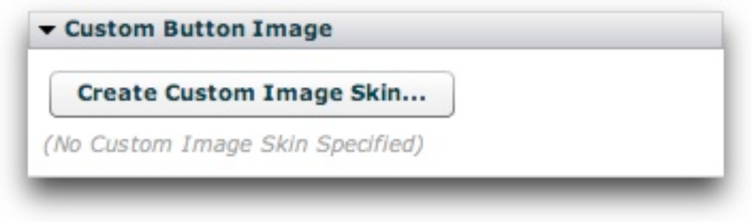


Again, note that all state images are exactly the same size. All state images for the selected-mode are placed at the bottom of the sprite image.

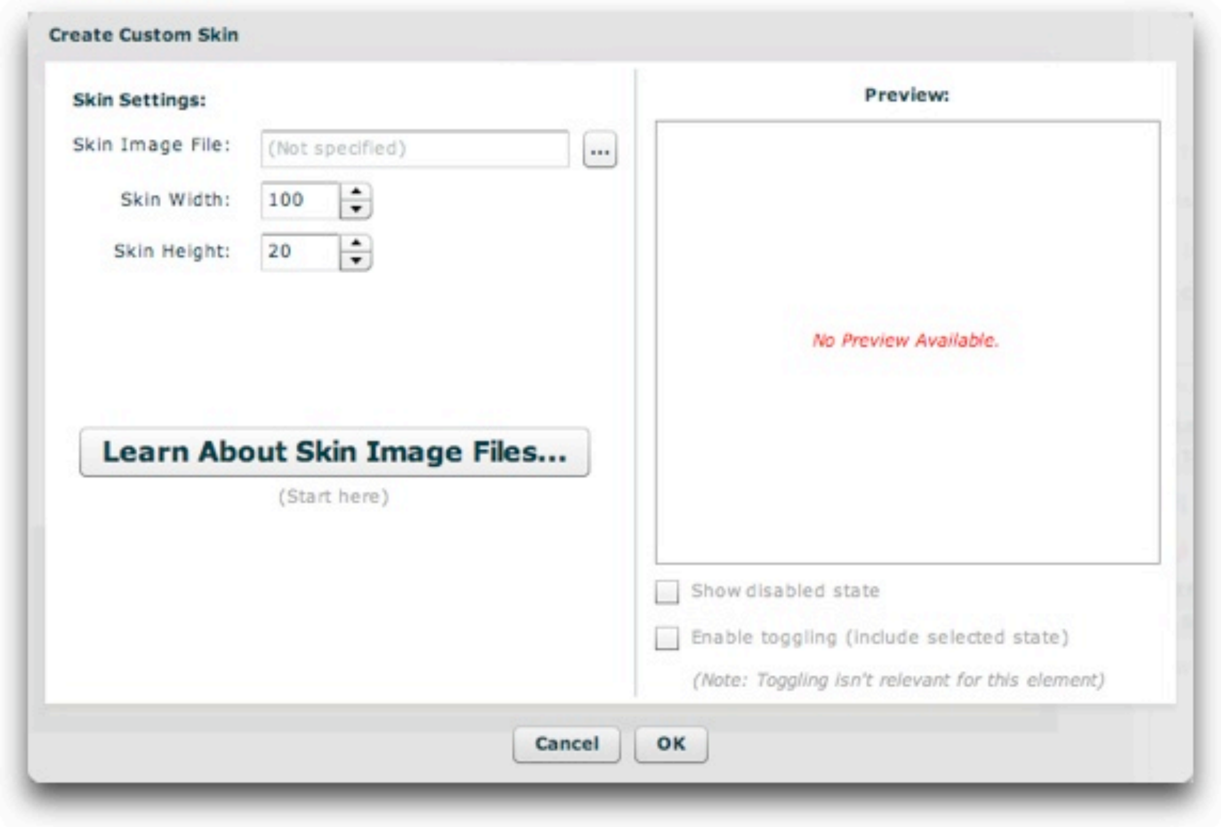
After you've created the sprite image in a graphics program of your choice (such as Photoshop or Fireworks), export the image as either a PNG, JPEG, GIF, or SWF file and save it within your site. It is generally best to use PNG 32 format, because it supports transparency and preserves high quality.

Applying a Custom Button Image in the Wizard

Once you have created your custom button sprite image, you're ready to apply it to your slideshow in the wizard. Launch the Insert or Edit Wizard and switch to the main Layout / Presentation tab. Locate and select the button element that you want to skin. In the Properties Panel, locate the Custom Button Image pod and click the Create Custom Image Skin button:

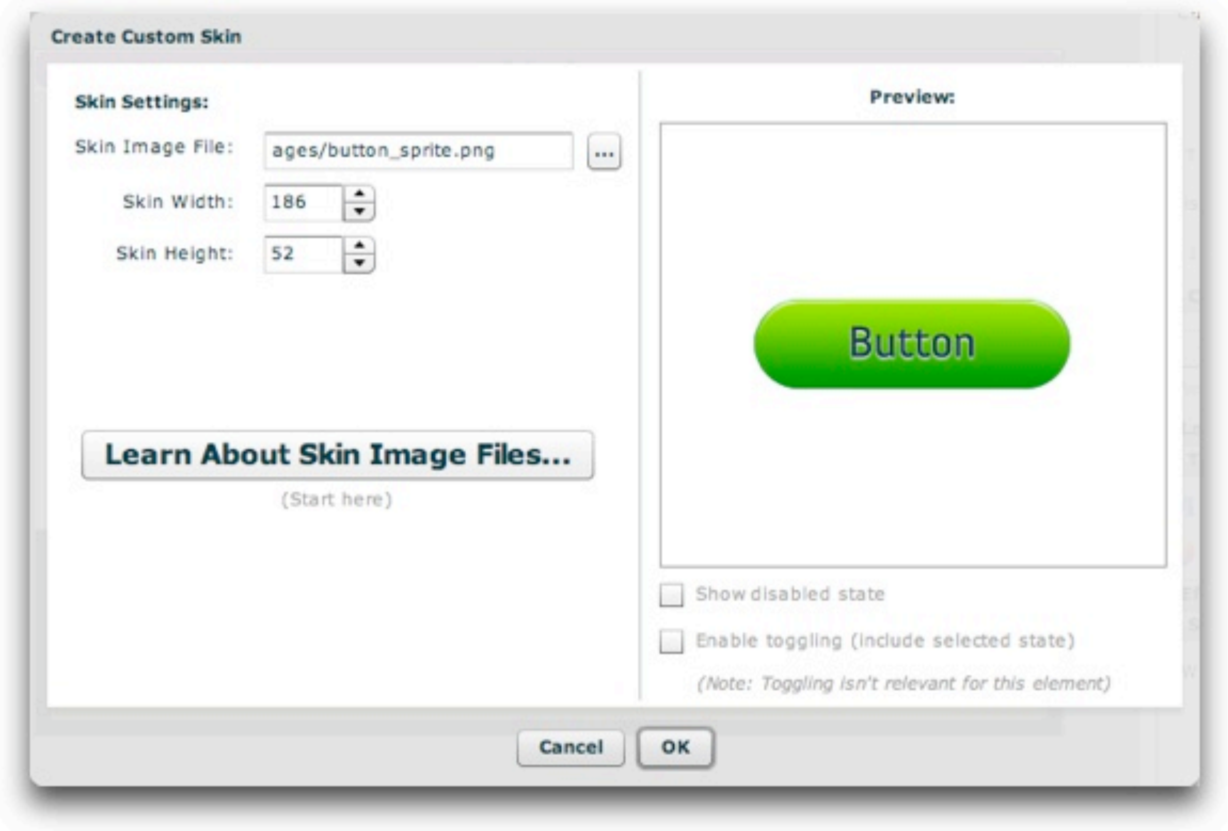


This will launch the following dialog:



On the left you can specify the the skin information; on the right a preview will be shown. Use the browse button on the Skin URL field to select the sprite image you just exported. After selecting an image, the Skin Width and Skin Height fields will be automatically filled. Make sure that the values are correct, corresponding to the dimensions of each state image within the sprite image (as shown in the diagrams in the section above).

On the right, you'll be able to see a preview of the button skin:



Create Custom Skin

Skin Settings:

Skin Image File: ages/button_sprite.png

Skin Width: 186

Skin Height: 52

Preview:

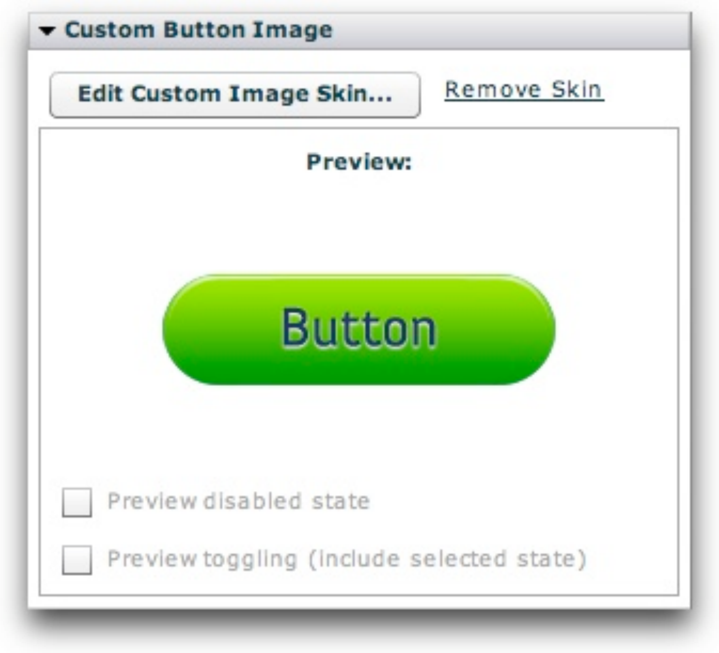
□ Show disabled state

□ Enable toggling (include selected state)

(Note: Toggling isn't relevant for this element)

At the bottom of the preview, there are two options that allow you to fully preview the button. First, the *Show Disabled State* checkbox allows you to preview the button in its disabled state. For two-mode buttons, the bottom option (*Enable Toggling*) will be checked by default, allowing you to preview both the standard and selected modes of the button. When this checkbox is selected, the example button will toggle back and forth as you click on it, so that you can see both button modes. For one-mode buttons, a message is displayed to let you know that the toggling is irrelevant for that element.

Once you're finished specifying the information and have confirmed that the button behaves as expected, click OK to apply the skin. You'll now be able to see a preview of the skin in the Properties Panel:


Custom Button Image
Edit Custom Image Skin...
Remove Skin
Preview:
Button
□ Preview disabled state
□ Preview toggling (include selected state)

Click the *Edit Custom Image Skin* button to return to the previous dialog to make changes. Click the *Remove Skin* button to remove the custom button skin. When you now preview your slideshow, the button will use the custom image as its skin.

Using Adobe Flash CS3 or CS4 or CS5+

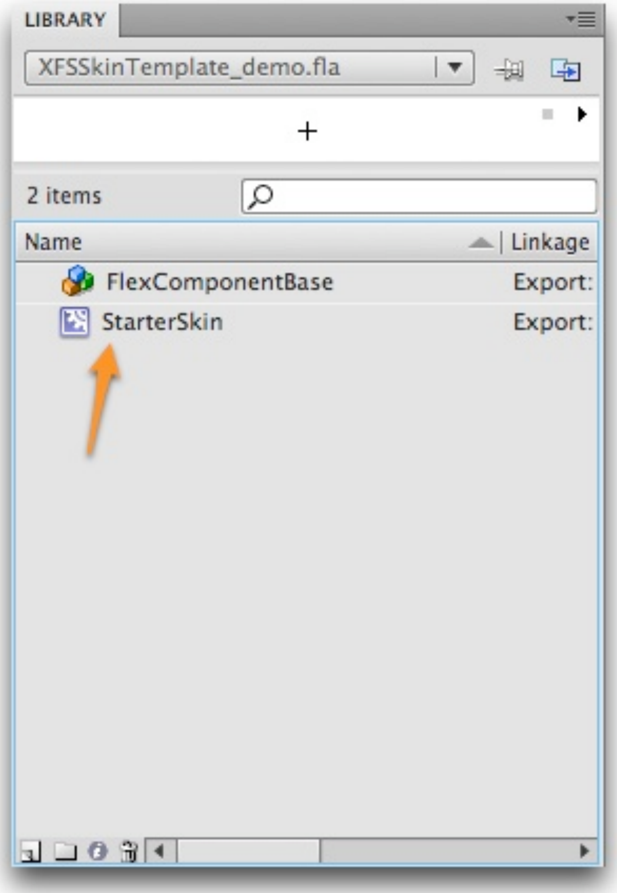
To use Flash to create skin packages, you must be using Adobe Flash CS3 or CS4 or CS5+. Additionally, you should have basic familiarity with Flash. To get started learning the basics of Flash, see the Flash Support page.

Please note: *If you just want to create customized buttons, it is much easier to use Easy Image Based Button Skinning.*

To begin, download the skin file starter FLA, available here. Save it to an easy to find location so it can be easily re-used when creating additional skin files. Place a copy of the starter FLA in a folder in the site you're working on, giving it a meaningful name such as `slideshow_skins.fla` .

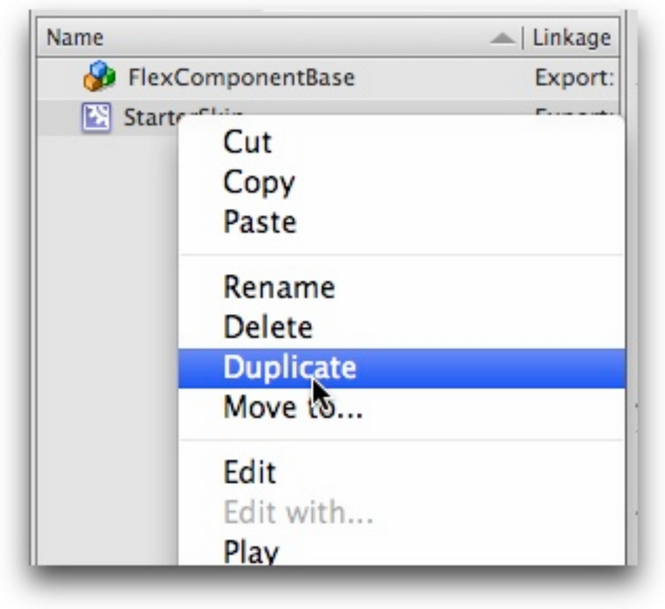
Important: It is highly recommended that you place the skin FLA in the `v4flashslideshow/` supporting files folder. While not required, this can save the trouble of later adding the skin file's folder as trusted content to enable previewing the custom skins from within the wizard.

Open the new copy of the file in Flash, and switch to the Library panel (Window > Library). You will see two assets, the `FlashComponentBase` compiled clip (which provides necessary integration infrastructure) and the `StarterSkin` movieclip:

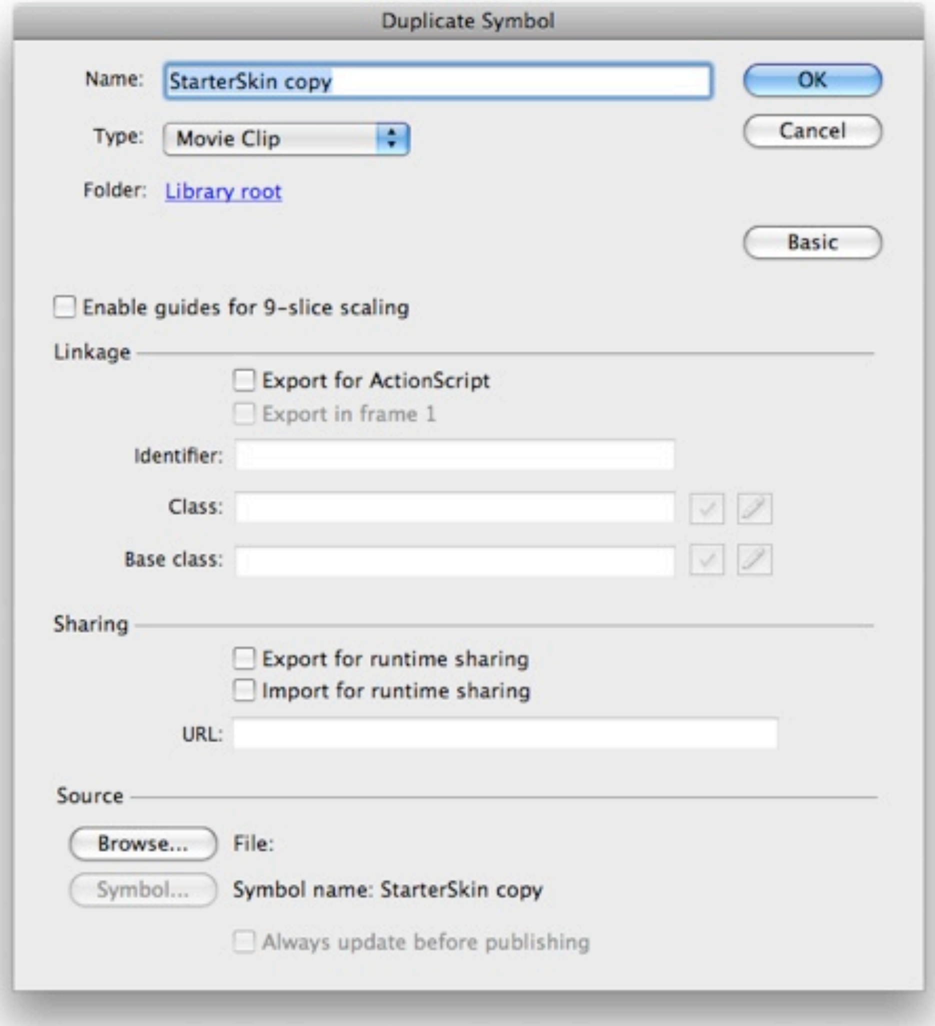


Name	Linkage
FlexComponentBase	Export:
StarterSkin	Export:

The StarterSkin movieclip is the base template for each of the skin(s) you create. It will never be used directly; instead, it will be duplicated for each skin you want to create. To start creating a new symbol, you will Cmd+click (right click) on the StarterSkin clip and select 'Duplicate':



This will open the Duplicate Symbol dialog:



Name: StarterSkin copy

Type: Movie Clip

Folder: Library root

□ Enable guides for 9-slice scaling

Linkage

□ Export for ActionScript

□ Export in frame 1

Identifier:

Class:

Base class:

Sharing

□ Export for runtime sharing

□ Import for runtime sharing

URL:

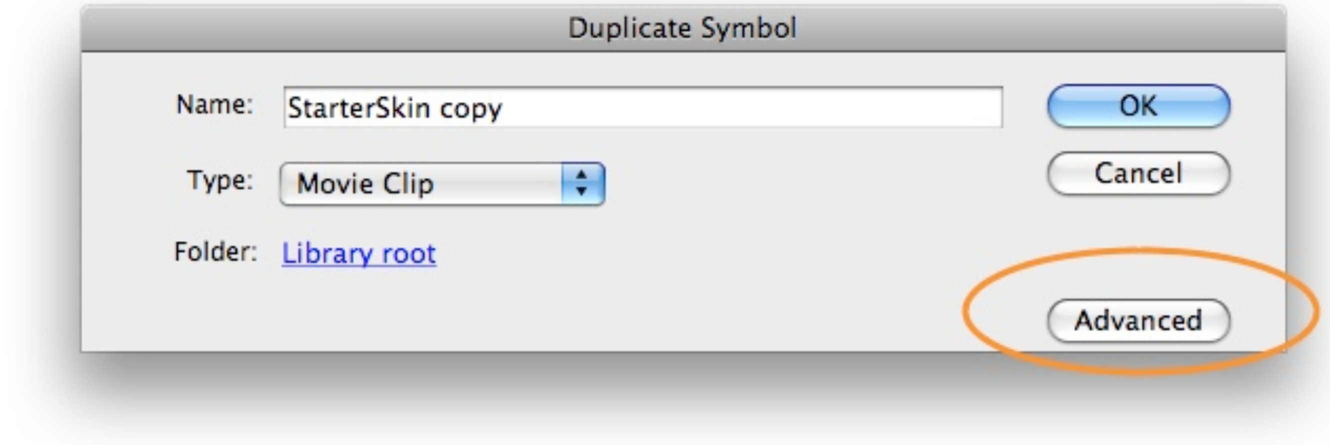
Source

File:

Symbol name: StarterSkin copy

□ Always update before publishing

Note: If you don't see all of the options listed above, click the Advanced button to expand the dialog:



Duplicate Symbol

Name: StarterSkin copy

Type: Movie Clip

Folder: Library root

Symbol Naming

First, you must enter a name for the symbol. The names are very important, as they determine which element(s) the skin or icon is automatically applied to, if any. *(Actually, the Class value does this, but this field is automatically filled when you enable Export for ActionScript, as described in a moment).* To create a skin which is automatically applied, enter the element name (e.g. `NextButton`) followed by `Skin` . So, to create a skin which is automatically applied to all NextButton element instances, enter a name of `NextButtonSkin` . If you want to create an icon which is automatically applied, append `Icon` to the element name. So, to create an icon which is automatically applied to all NextButton instances, enter a name of `NextButtonIcon` . The following table summarizes the elements for which custom skins are commonly used:

Element Name	**Selected State Applies?** ***(more on this below)***	**Notes**
PrintButton	No	
FirstButton	No	
LastButton	No	
PlayPauseButton	**Yes**	The standard (unselected) state is where the 'Play' button image will go. The selected state is where the 'Pause' button image will go.
PreviousButton	No	
NextButton	No	

LayoutJumpButton	No	
FullScreenButton	**Yes**	The standard (unselected) state is where the 'Go full screen' button image will go. The selected state is where the 'Return to normal screen' button image will go.
DynamicLinkButton	No	
LinkButton	No	
Button	*Usually, no.*	The selected state only applies if you are using the toggle feature on the button.
AutoplayToggleButton	**Yes**	The standard (unselected) state is where the 'Play' button image will go. The selected state is where the 'Pause' button image will go.
CaptionToggleButton	**Yes**	The standard (unselected) state is where the 'Caption currently visible' button image will go. The selected state is where the 'Caption currently hidden' button image will go.
AutoplayLoopToggleButton	**Yes**	The standard (unselected) state is where the 'Loop currently disabled' button image will go. The selected state is where the 'Loop currenty enabled' button image will go.
AudioMuteButton	No	
AudioVolumeToggleButton	**Yes**	The standard (unselected) state is where the 'Sound is on' button image will go. The selected state is where the 'Sound is muted' button image will go.
AudioPlayPauseButton	**Yes**	The standard (unselected) state is where the 'Play' button image will go. The selected state is where the 'Pause' button image will go.
AudioLoudButton	No	

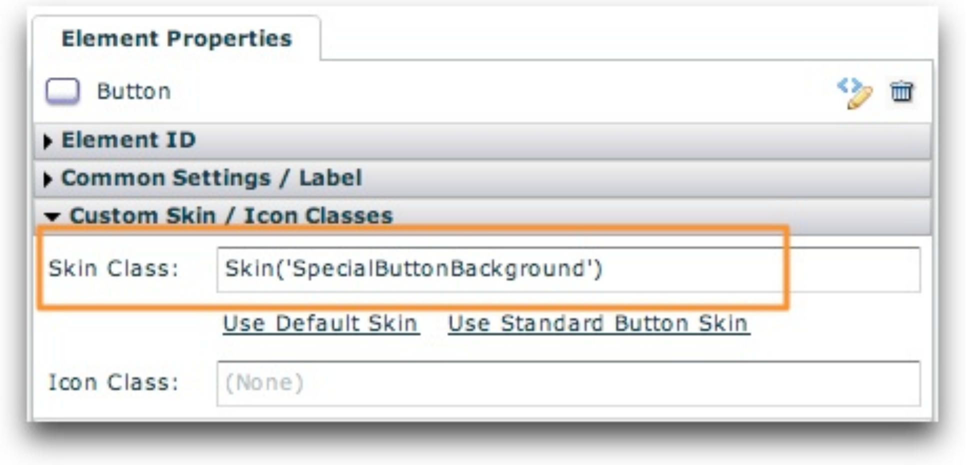
For information on all elements in this table, see the Available Elements section of the manual.

Some buttons toggle, giving them two separate states (e.g. the PlayPauseButton element). On these elements, there are separate skins which are specified for the second state. Using the example of the PlayPauseButton, the standard state is where the Play button goes, and the selected state is where the Pause button goes. However, on elements which only have one state, no skins need be specified for the selected state.

Tip for Advanced Users: Instead of using an element's name as the base (the value which precedes Skin or Icon), you can also use some other generic name. In this case, the skin element will be applied to all elements which have a ... styleName="***baseName***" ... attribute on them. For example, if you created a skin symbol named specialButtonSkin (a base value of specialButton) it would be applied to any Button (or other type of) element which looked like this:

```
<Button ... styleName="specialButton" ... />
Or:
<SomeOtherElement ... styleName="specialButton" ... />
```

If you use a symbol name which doesn't end in Skin or Icon, the skin won't be automatically applied to any elements. Instead, you'll be able to reference it in the slideshow by using the `Skin('`***`MySymbolName`***`')` attribute format. For example, you might use multiple generic Button elements in your slideshow. If you wanted to apply a customized skin to just one of these instances, you could create a symbol named `SpecialButtonBackground`. Then, in the Properties Panel for the appropriate Button element, you would enter `Skin('SpecialButtonBackground')` in the Skin field:

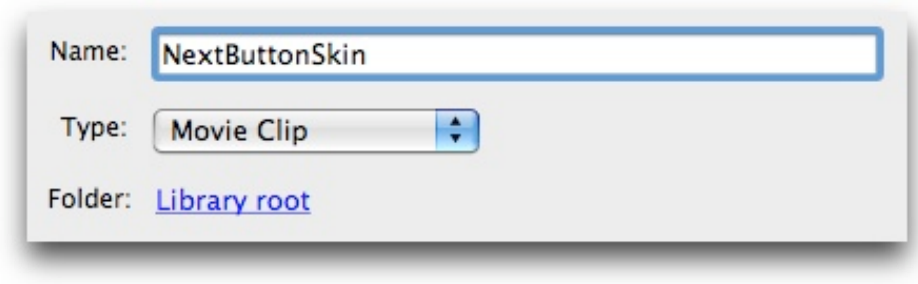


Skin Class: Skin('SpecialButtonBackground')

Icon Class: (None)

While icons can be used on button elements, it is generally much better to use a full customized skin instead of an icon. Icons are overlaid on top of the skin, whereas a skin represents the entire button's appearance.

For this example, we will be creating a customized skin for the NextButton element by using `NextButtonSkin` as the name:



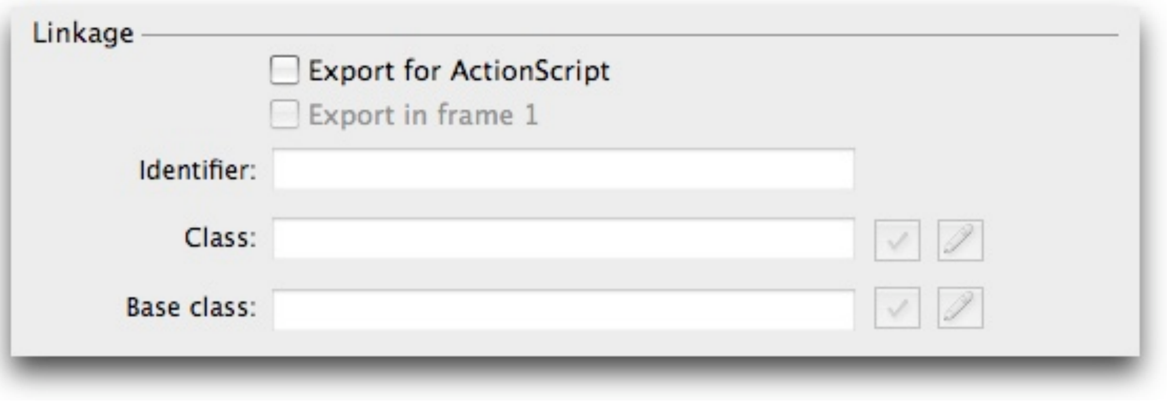
Name: NextButtonSkin

Type: Movie Clip

Folder: Library root

Symbol Linkage

Once you have entered a symbol name in the Duplicate Symbol dialog, you must configure the Linkage section as well:



Linkage

□ Export for ActionScript

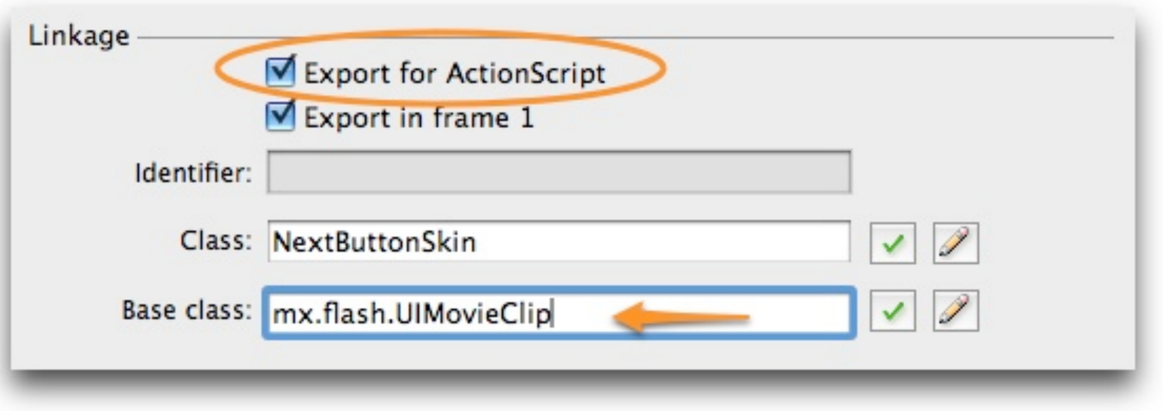
□ Export in frame 1

Identifier:

Class:

Base class:

First, check the 'Export for ActionScript' option. This will enable and pre-populate the 'Export in Frame 1', Class and Base Class fields. **Important:** You must change the Base Class field to be `mx.flash.UIMovieClip` or the skins won't work. The Class box will be automatically filled with the symbol name you entered above; make sure that this **exactly matches** the symbol name, or the skins won't work. After updating the settings, the section should appear like this:



Linkage

☑ Export for ActionScript

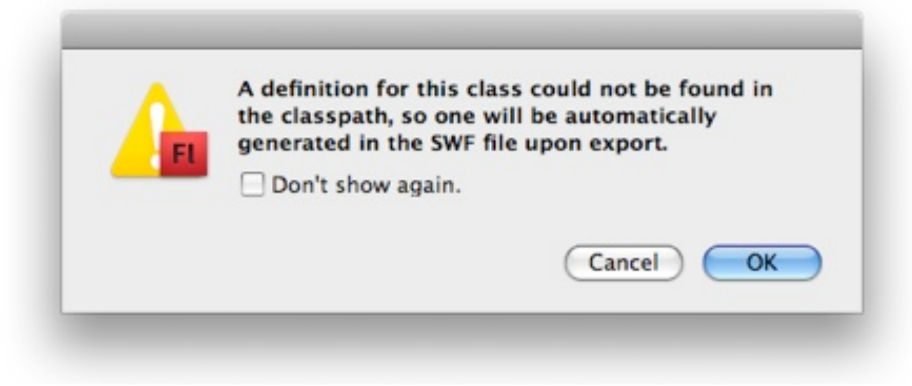
☑ Export in frame 1

Identifier:

Class: NextButtonSkin

Base class: mx.flash.UIMovieClip

You can now click the OK button on the right of the Dupliate Symbol dialog to finish duplicating the symbol. You may receive the following warning message; if so, simply click OK:

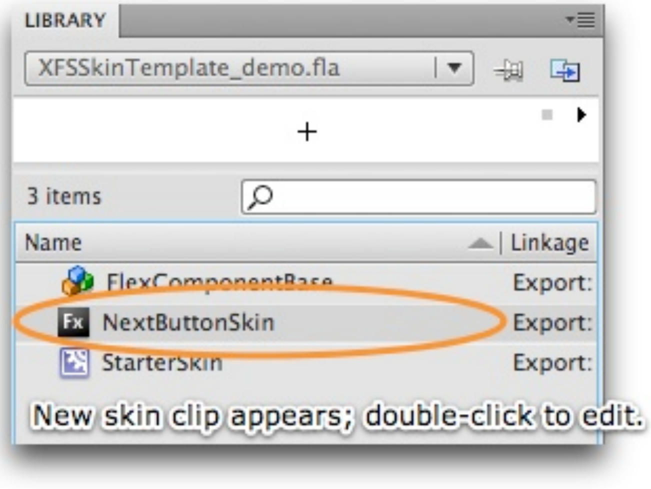


A definition for this class could not be found in the classpath, so one will be automatically generated in the SWF file upon export.

□ Don't show again.

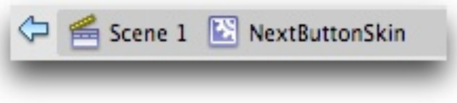
Symbol Editing

The newly created skin symbol will now appear in your Library panel:

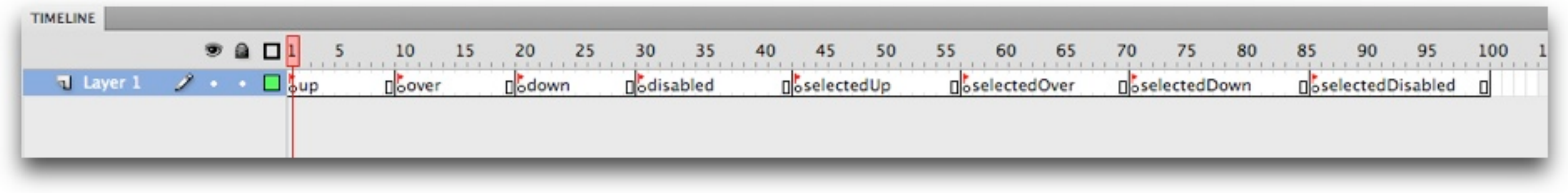


New skin clip appears; double-click to edit.

To begin editing the skin, double-click it. Above the stage, you should see that you're editing the NextButtonSkin movieclip symbol (for this demo):



With the symbol now selected for editing, switch to the Timeline (Window > Timeline):



Each of the labeled keyframes will hold a different state of the button. To edit a state, click on its keyframe (the circle just below the red flag), then add the desired contents to the stage. (If the process of editing a movieclip if unfamiliar to you, you need to review the Flash help.) The following table summarizes what button state each of the frames represents:

State	Info
	States which apply to all buttons/elements:
up	Represents the skin when there is no user interaction. This is its default/normal state.
over	Represents the skin when the user hovers over it. (E.g. when they roll over a button.)
down	Represents the skin when the user presses down the mouse on it. (E.g. when they click the button.)
disabled	Represents the skin when the element (button) is disabled. (E.g. the PreviousButton is disabled when the first image is being displayed.)
	States which only apply to toggled (selectable) buttons/elements:
selectedUp	*Only applies to the elements listed in the table above which utilize the selected state.* Represents the up state (as described above) when the button is selected.
selectedOver	*Only applies to the elements listed in the table above which utilize the selected state.* Represents the over state (as described above) when the button is selected.
selectedDown	*Only applies to the elements listed in the table above which utilize the selected state.* Represents the down state (as described above) when the button is selected.
selectedDisabled	*Only applies to the elements listed in the table above which utilize the selected state.* Represents the disabled state (as described above) when the button is selected.

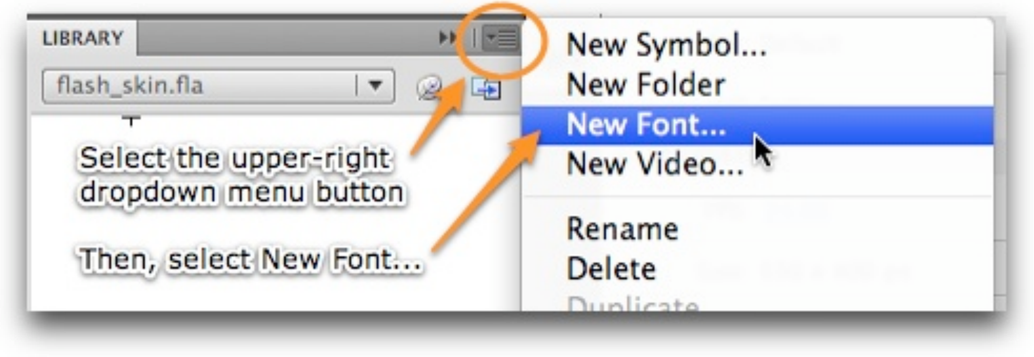
As noted in the table, elements with only a single state (e.g. the NextButton) need not have any content added to the final four keyframes. Additionally, note that all of the specific frame label names are important -- don't modify them, or the skins won't work properly.

Once you have added content to each of the frames, the skin symbol is complete. If you wish to create additional skins, follow the process of duplicating the

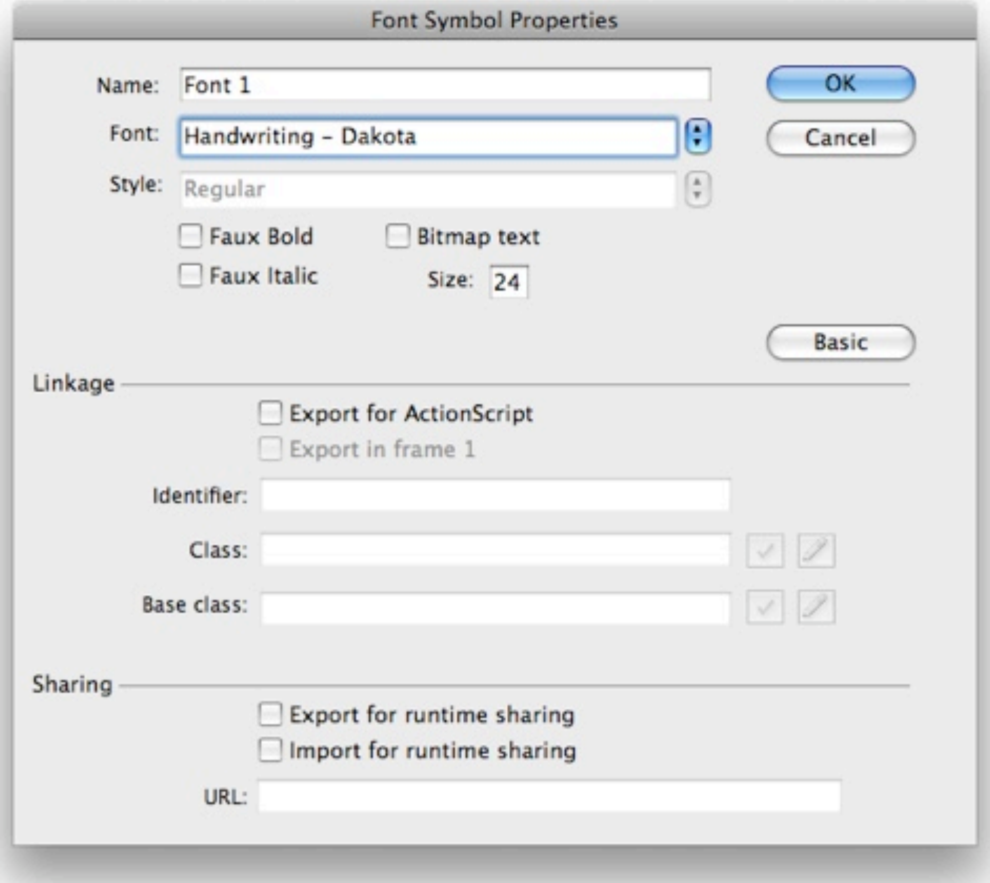
`StarterSkin` template again.

Embedding Fonts

Skin files can also be used to hold embedded fonts for use in the slideshow. Open the Library panel and select the dropdown menu in the upper-right corner of the panel. On the menu, select 'New Font...':



This will open the New Font Symbol dialog:



Font Symbol Properties

Name: Font 1

Font: Handwriting – Dakota

Style: Regular

□ Faux Bold □ Bitmap text

□ Faux Italic Size: 24

Linkage

□ Export for ActionScript

□ Export in frame 1

Identifier:

Class:

Base class:

Sharing

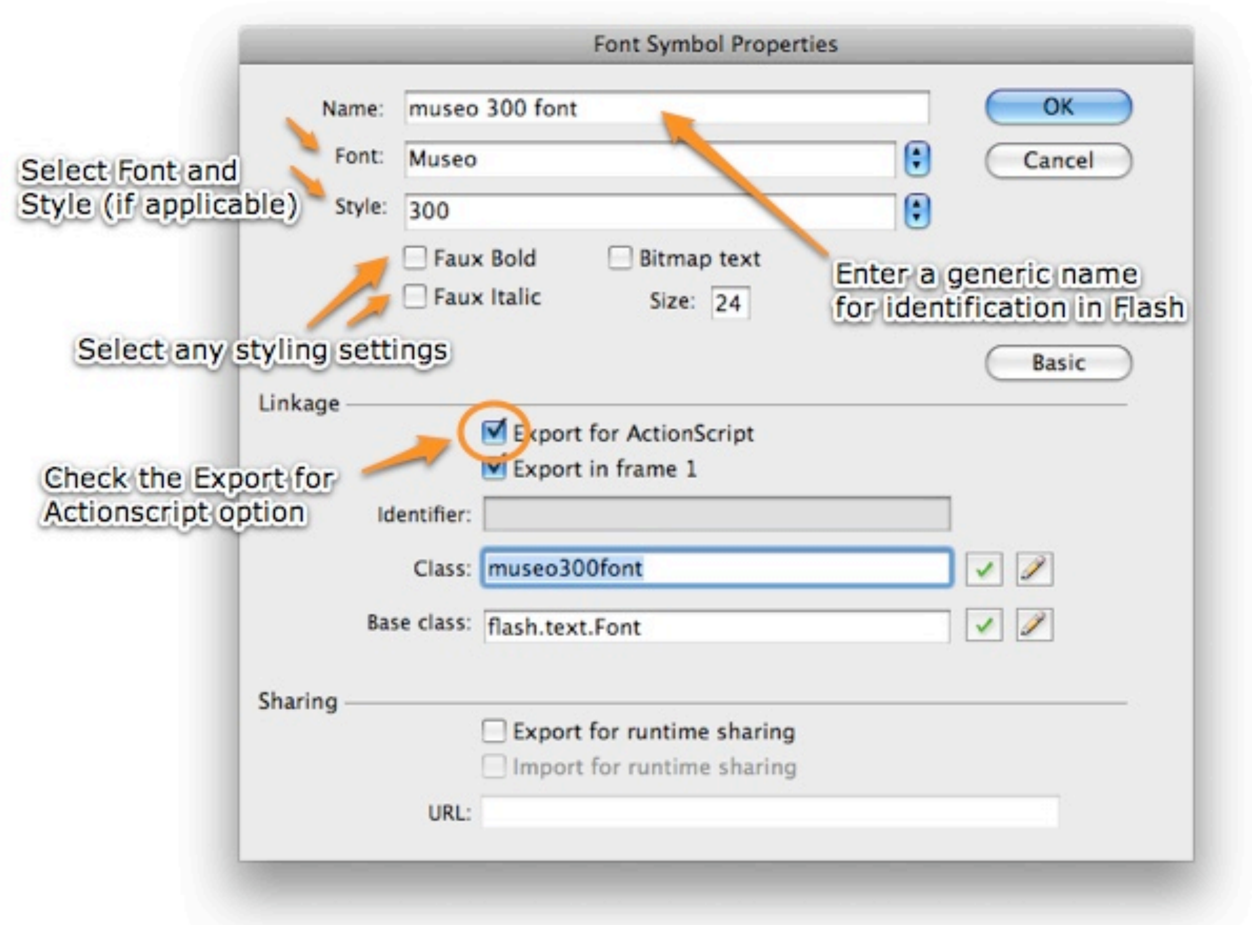
□ Export for runtime sharing

□ Import for runtime sharing

URL:

First, select a font from the Font menu. If the font has multiple styles, you will be able to select one from the Style menu. If you want to embed a bold, italic, or bold+italic version of the font, check the appropriate boxes. If you plan to use multiple styles of the font in your slideshow, you must embed each style of the font as a separate symbol. Next, enter a generic name in the Name field -- this name is only used to identify the font when working in Flash; it will not be referenced in the actual slideshow.

Next, you must check the Export for ActionScript checkbox. This step is very important! When you check the box, the Export in Frame 1 box will be automatically checked as well. The dialog should now appear similar to the following:



Font Symbol Properties

Name: museo 300 font

Font: Museo

Style: 300

□ Faux Bold □ Bitmap text

□ Faux Italic Size: 24

Linkage

☑ Export for ActionScript

☑ Export in frame 1

Identifier:

Class: museo300font

Base class: flash.text.Font

Sharing

□ Export for runtime sharing

□ Import for runtime sharing

URL:

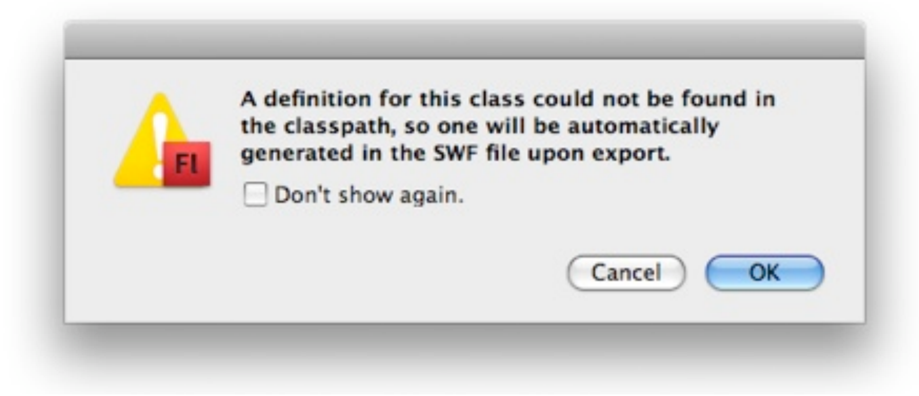
Select Font and Style (if applicable)

Enter a generic name for identification in Flash

Select any styling settings

Check the Export for Actionscript option

Once you've configured the settings, click the OK button to add the font symbol to the FLA's library. You may receive the following warning message; it can be ignored:



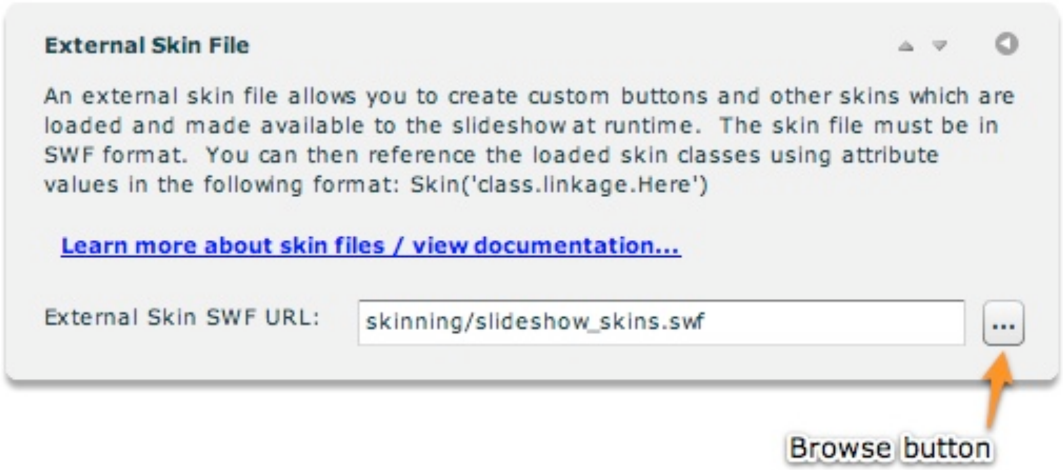
A definition for this class could not be found in the classpath, so one will be automatically generated in the SWF file upon export.

□ Don't show again.

Once you have exported and applied the skin file to your slideshow, see the Using Custom Embedded Fonts topic for more information on utilizing the embedded font(s) in your slideshow.

Exporting and Applying the Skin Package

If you saved the skin FLA to a location within your site, the exporting process is very straightforward. Simply press Cntrl + Enter (Cmd + Return) to test the movie. (You can also select Control > Test Movie.) This will generate the SWF file skin package and place it in the same folder as the FLA. Now, switch to Dreamweaver and launch the slideshow wizard. In the External Skin File pod in the Appearance Settings tab, browse to and select the SWF file which was just generated:



External Skin File

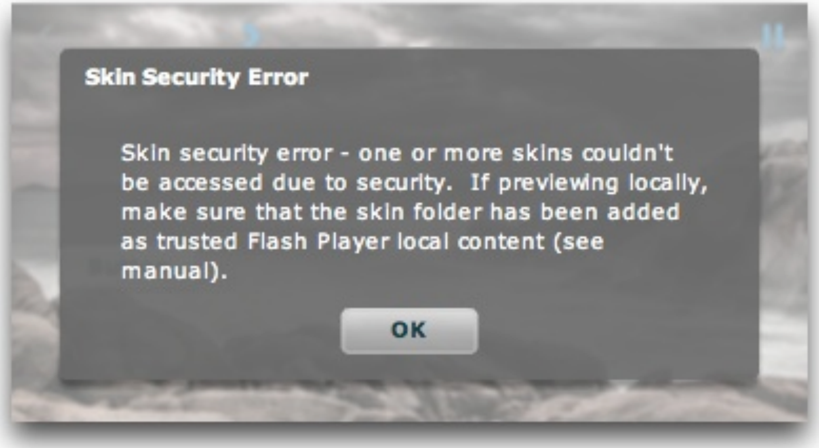
An external skin file allows you to create custom buttons and other skins which are loaded and made available to the slideshow at runtime. The skin file must be in SWF format. You can then reference the loaded skin classes using attribute values in the following format: Skin('class.linkage.Here')

Learn more about skin files / view documentation...

External Skin SWF URL: skinning/slideshow_skins.swf

When you save and preview the slideshow, all of your skins should be automatically applied:

IMPORTANT: If you didn't place the skin file in the already-trusted `v4flashslideshow/` supporting files directory, you may receive the following error message when previewing the slideshow:



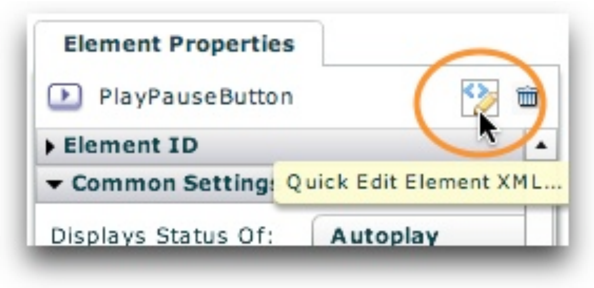
Skin Security Error

Skin security error - one or more skins couldn't be accessed due to security. If previewing locally, make sure that the skin folder has been added as trusted Flash Player local content (see manual).

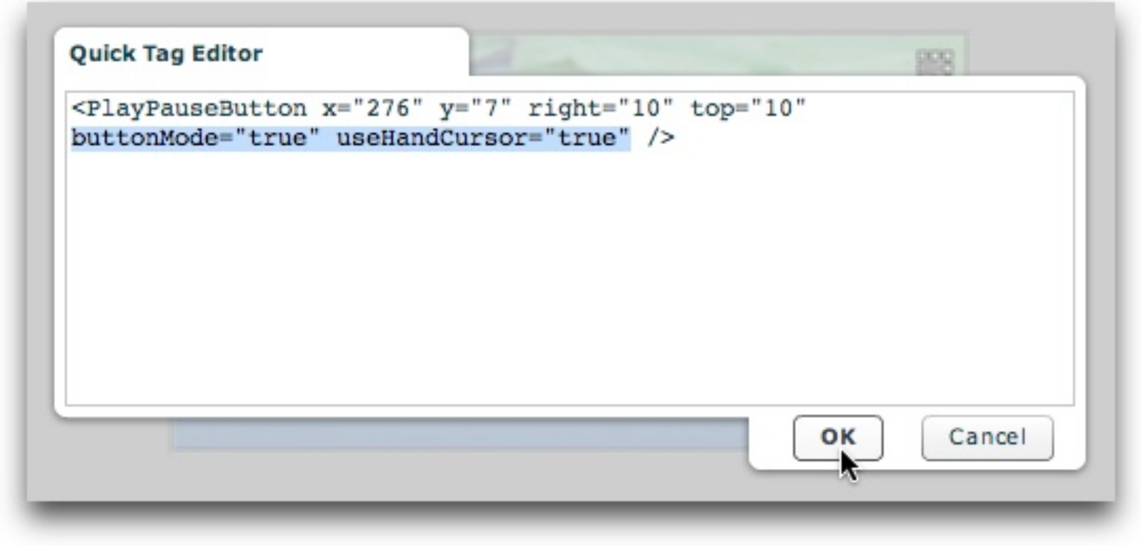
To address this issue, you must add the folder which holds the skin SWF as Flash Player trusted local content in the Flash Player security settings dialog. To learn more about doing this, see this topic.

General Tips

If you want any button (especially one with a custom skin) to display the hand cursor when moused over to help make its action more clear to users, you can enable this behavior by adding `...buttonMode="true" useHandCursor="true"...` to the button's element code. In the wizard, select a button and click the Quick Edit Element XML button:



In the Quick Tag Editor dialog, add the code shown above (highlighted in the picture):



```
<PlayPauseButton x="276" y="7" right="10" top="10"
buttonMode="true" useHandCursor="true" />
```

Click OK, and your button will now display a hand cursor when hovered over.

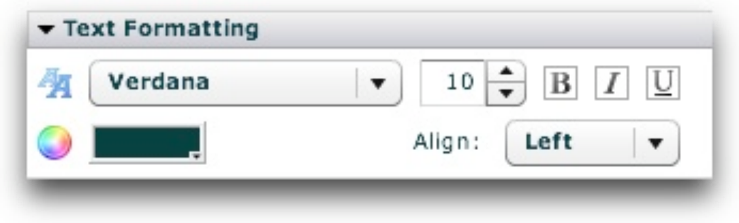
Advanced Topics / Advanced Examples > **Using Custom Embedded Fonts**

Using Custom Embedded Fonts

To utilize embedded fonts in a slideshow, you must first create a skin file and the embed the font(s) in the skin file. To learn about creating a skin file, see this topic, especially the Embedding Fonts section. Once you have created and applied the skin file as described in that topic, you're ready to use them in your slideshow.

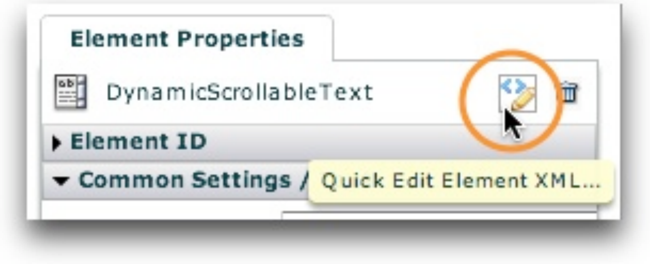
Using Embedded Fonts in a Slideshow

To reduce confusion, the font pickers in the wizard only allow selections from a list of web-safe fonts. In order to use an embedded font, you must make a small modification to the appropriate element code. First, locate the element you wish to customize and select it. If a font setting is applicable for this element, you'll see the Text Formatting section in the Properties Panel:



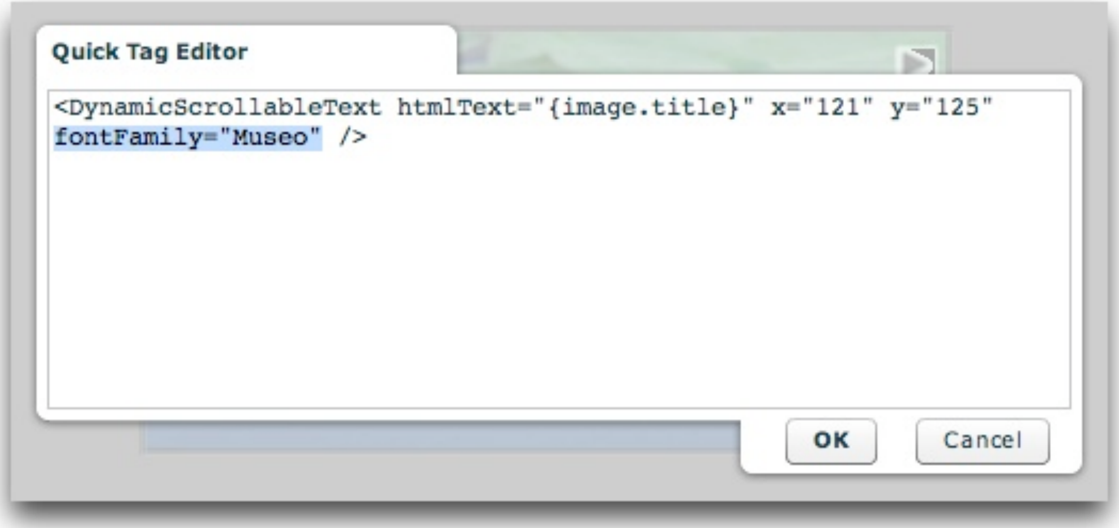
Verdana
10
Align: Left

Elements commonly customized to use embedded fonts include the DynamicText and DynamicScrollableText and DynamicLinkButton elements, since they often contain the image caption and title. With the element selected, click the Quick Edit Element XML button at the top of the Properties Panel:



This will launch the Quick Tag Editor window. In the element's XML, add `fontFamily="`***`MyFontName`***`"` , replacing ***MyFontName*** with the **actual name of**

the embedded font (e.g. *Museo* or *Trajan Pro*). Note that this name is NOT the name value you assigned to the font when embedding it in the skin file; it is the actual font name. The necessary change is shown with the highlighted code in the following screenshot:



```
<DynamicScrollableText htmlText="{image.title}" x="121" y="125"
fontFamily="Museo" />
```

Click OK to save your changes, and preview to make sure everything appears properly.

Tip: It is highly recommended that you test-view slideshows which include embedded fonts on a machine which has no special fonts installed to verify that the embedding took place properly.

Note: Embedded fonts will appear as Times in the wizard; you must preview to see the embedded font properly applied.

Note: When using an embedded font, make sure that the bold/italic settings on the slideshow element match the font embed settings used in Flash. For example, if you embedded the regular typeface in Flash (bold and italic options *disabled* in the Font Symbol Properties dialog), make sure that you don't use the bold or italic options on the slideshow element. If you want to use multiple versions of a font (e.g. some regular and others bold or italic), you must embed multiple versions of the font with corresponding formatting settings in the Font Symbol Properties dialog. Failing to do this properly can cause users to see the default font (Times) when the slideshow is deployed, often creating a disruptive visual experience.

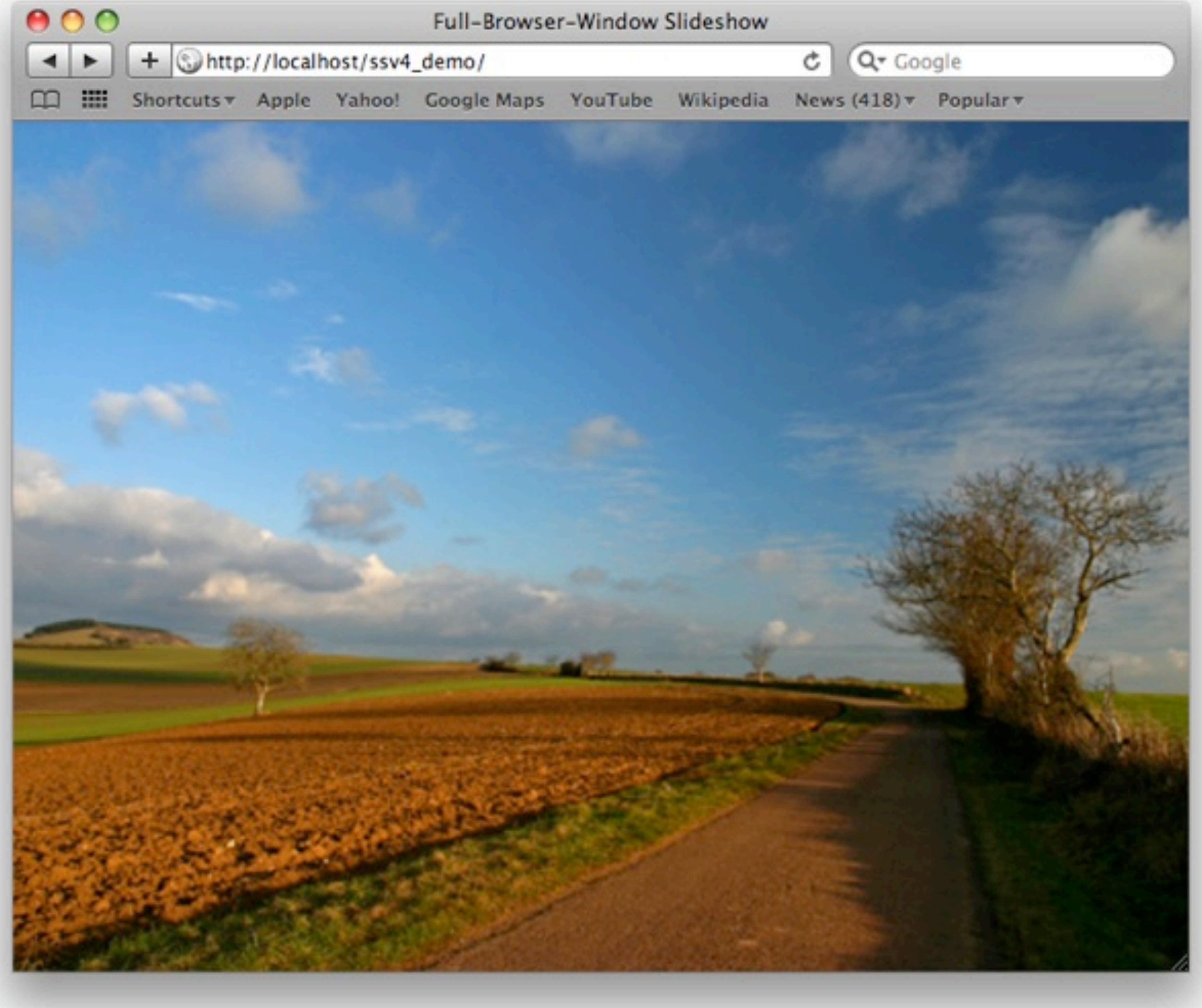
The Slideshow HTML Structure

To learn about how the Slideshow HTML is constructed, see the Overview of Slideshow HTML Structure in the Developer Reference.

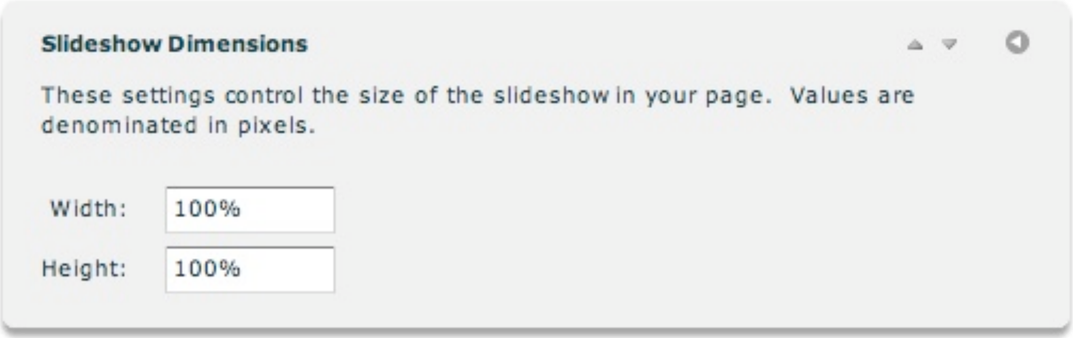
Creating a Full-Browser-Window Slideshow

This topic details the process of creating a slideshow which expands to fill the entire browser window. Note that this is different than true full-screen, as implemented through the FullscreenButton element.

Here is an example of a full-browser-window slideshow:



To create this type of implementation, first set the Slideshow Dimensions in the main Appearance Settings section to `100%` :



Slideshow Dimensions

These settings control the size of the slideshow in your page. Values are denominated in pixels.

Width: 100%

Height: 100%

Note: You can also set the dimensions in the Properties panel, using the Slideshow Inspector.

Next, save/exit the wizard and switch to Code View for your HTML page. In the `<head>` add the following code:

```
<style type="text/css">
body { padding: 0; margin: 0; overflow:hidden; height: 100%; width: 100%; }
</style>
```

Next, locate the slideshow HTML code block. Directly above the start of the slideshow HTML, add the bolded line of code:

```
<div style="position: absolute; left: 0; top: 0; width: 100%; height: 100%;">
<!-- Start dwuser_XML_Flash_Slideshow_v4 -->
```

Directly below the end of the slideshow HTML, add the bolded line of code:

```
<!-- End dwuser_XML_Flash_Slideshow_v4 -->
</div>
```

The slideshow will now expand to fill the entire browser window.

Deploying to a Blog or CMS

In some cases you may want to deploy a slideshow to a webpage overwhich you don't have direct control, such as a blog or a CMS. In these cases, you don't have the luxury of directly opening the page in Dreamweaver and inserting a slideshow. The solution to these situations is to create a separate page to display the slideshow, then display this page in your main page using an inline frame (`<iframe>`).

To do this, first create an HTML page in Dreamweaver (e.g. `slideshow_holder.html`) which is specifically designed to hold the slideshow. In the <head> of the holder page add the following code:

```
<style type="text/css">
body, html { margin: 0; padding: 0; }
</style>
```

Follow the standard instructions for inserting a slideshow into this page. Deploy it as detailed in that section of the manual, uploading all of the necessary files along with the holder page. After uploading the page, determine the full path to the slideshow page -- e.g. `http://www.example.org/slideshows/slideshow_holder.html`.

Next, in your other page (e.g. the blog or CMS), add the following code to display the slideshow:

```
<iframe scrolling="no" frameborder="0" width="500" height="300"
src="http://www.example.org/slideshows/slideshow_holder.html"></iframe>
```

You must replace the bolded `src` value with the slideshow holder page URL you determined earlier. The bolded `width` and `height` values must be replaced with the slideshow's actual dimensions.

Color Correction

General Background / Overview

Note: The information in this topic may evolve over time. All information was accurate at time of writing.

Color correction is used to ensure that image colors appear the same to all users, regardless of differences in the medium (usually a monitor) used to view that image. Currently, most web browsers don't support color correction. The only two browsers which do support color correction are Safari 3+ (under 9% penetration at time of writing), and Firefox 3+. However, Firefox does **not** have color correction enabled by default. As a result, the vast majority of web users view all web content with color uncorrected. **Update:** Now Firefox has color correction for some photos enabled by default -- see http://hacks.mozilla.org/2009/06/color-correction/ .

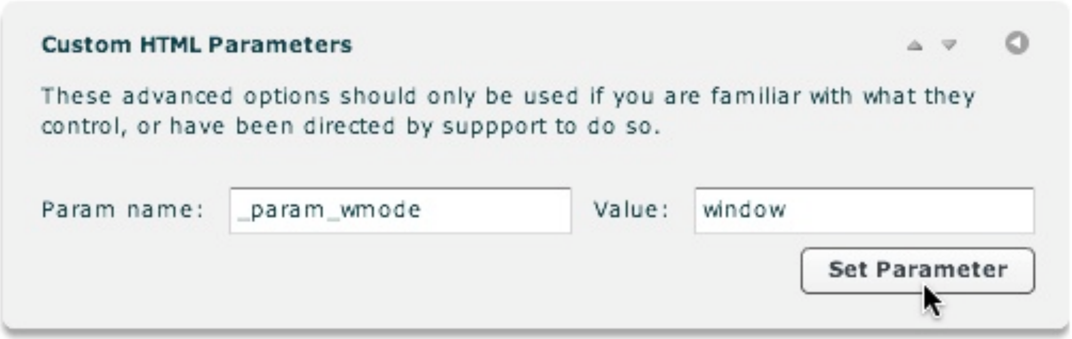
When Flash content (such as an XFS v4 slideshow) is included in a web page, it plays in its own "island" of sorts. Until recently, this meant that Flash content never had any color correction, period. If colors used in a SWF were supposed to match the HTML page, they would 90%+ of the time. Only in browsers with color correction enabled, such as Safari, would the colors not match due to correction being only applied to the HTML-based colors. With the introduction of the Flash Player 10, however, native color correction support was included in the Flash Player. The intended behavior is for color correction to only be enabled if the host browser has color correction enabled, always ensuring a color match with any HTML colors. However, at the time of writing, this behavior is buggy at best. In reality, Flash Player color correction is never or almost never enabled by default, even if the host browser has color correction enabled (most notably in Safari 3+). While the default "matching" behavior doesn't function properly, it still is possible to explicitly enable color correction for content playing in the Flash Player (ignoring the host browser's status). **This will ensure preservation of the color fidelity of all images.** The only downside is that colors in the slideshow may not match corresponding colors in the HTML page, because the 90%+ of users without browser-based color correction will see uncorrected colors in the HTML. You can either not attempt to have matching colors in the slideshow, or use Flash (with color correction explicitly enabled) to display your other page content.

Using Color Correction in XFS v4 Slideshows

(Make sure you've read the topic above before reading this section!) When exporting images for use with slideshows, it is best to always use sRGB for both the display profile and image profile.

To use color correction, you must be using the Professional version of the software. Learn about upgrading here.

To enable any type of color correction, you must first set the `wmode` SWF embed parameter to be `window` instead of the default, `transparent`. The downside of using `window` as the `wmode` is that the slideshow background will no longer be transparent; if you wish to fake a transparent background, you'll have to include a static image in the slideshow which looks like the page background that would otherwise show through. To apply this setting, you'll use the Custom HTML Parameters pod in the Global Configuration and Playback Settings section, with a parameter name of `_param_wmode` :

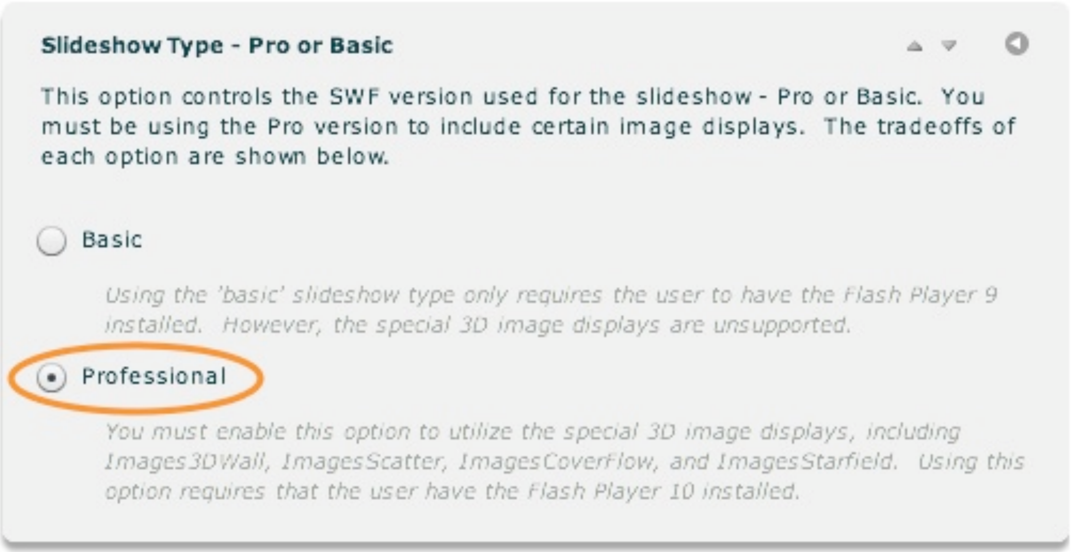


Custom HTML Parameters

These advanced options should only be used if you are familiar with what they control, or have been directed by suppport to do so.

Param name: _param_wmode Value: window

Next, in the 'Slideshow Type - Basic or Pro' pod of the Global Configuration and Playback Settings section, you must select the Professional SWF version:



Slideshow Type - Pro or Basic

This option controls the SWF version used for the slideshow - Pro or Basic. You must be using the Pro version to include certain image displays. The tradeoffs of each option are shown below.

○ Basic

Using the 'basic' slideshow type only requires the user to have the Flash Player 9 installed. However, the special 3D image displays are unsupported.

● Professional

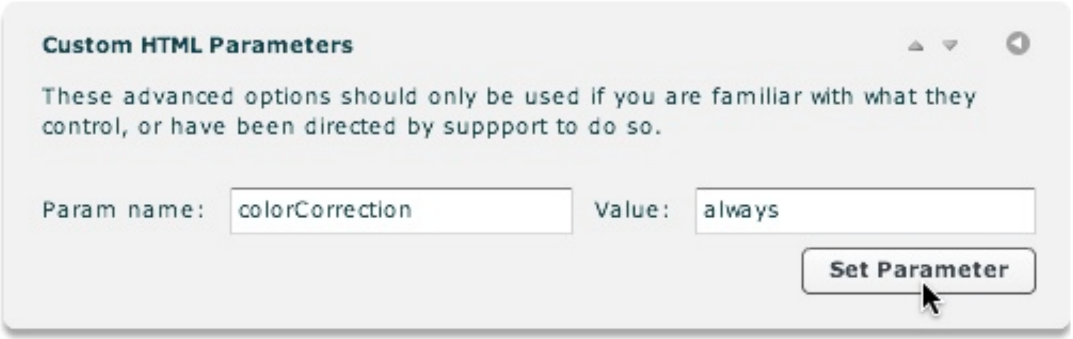
You must enable this option to utilize the special 3D image displays, including Images3DWall, ImagesScatter, ImagesCoverFlow, and ImagesStarfield. Using this option requires that the user have the Flash Player 10 installed.

Finally, you need to set the `colorCorrection` parameter (documented in the `com.dwuser.ss4.core.FlashSlideshowConfigurationParameters` class in the API Reference). The following settings are available:

- **`match`** - (Default) Leaves all color correction decisions up to the Flash Player. As mentioned above, in theory this is supposed to make the slideshow match the browser.

- **forceMatch** - Attempts to determine if color correction is enabled in the browser. If so, slideshow color correction is enabled. At the time of writing, this fails due to Flash Player bugs.
- **always** - Always enables color correction, regardless of the host browser's setting. This means the fidelity of all images will always be preserved. However, the slideshow may not match the host page properly.

At the time of writing, the only relevant value to use is the `always` setting. As mentioned above, this will ensure that the color fidelity of your images is always preserved. This setting, like the `wmode` setting, is applied in the Custom HTML Parameters pod of the Global Configuration and Playback Settings section. The parameter name will be `colorCorrection` and the value `always` :



Custom HTML Parameters

These advanced options should only be used if you are familiar with what they control, or have been directed by suppport to do so.

Param name: colorCorrection Value: always

After saving changes, the slideshow should appear with color correction applied. Note that the entire slideshow as a whole will have color correction applied (not just the images).

Using Permalinks / Linking Directly to Galleries and Images

Permalinks are a special type of link which point to a specific item or area of a page. In the case of a slideshow, permalinks are used to link directly to specific gallery or even a specific image within a gallery. This topic explains how to use permalinks to link directly to specific galleries or images in a slideshow.

The UniqueID Format

First, it is important to understand the format used to reference specific galleries and images. There are two ways to reference a gallery or an image:

- An image or gallery can be references using its zero-base numeric index. Thus, if you wanted to reference the third gallery in a slideshow, its numeric index would be **2**
- An image or gallery can be referenced by its unique ID value, as specified in the Gallery Properties or Image Properties dialog. For galleries, this value is the Linkage ID value in the General Settings tab; for images, this value is the Image ID value in the Title / Description / Tooltip tab.

There are two different types of reference strings / UniqueID's -- a reference just to a specific gallery, and a reference to a specific image in a specific gallery. The format for these references is as follows:

- Reference to a single gallery: The gallery's ID or numeric index is used; nothing else is needed. For example, to reference the third gallery in a slideshow which has an ID value of `specialGallery` , you could use either of these two formats:
    - `2`
    - `specialGallery`

- Reference to a single image in a specific gallery: The gallery's ID or numeric index comes first, followed by `->` and the image's ID or numeric index. Here are some examples:
    - To reference the third image in the second gallery using numeric indices, this reference string would be used: **1->2**
    - To reference an image with an ID of `specialImage` in a gallery with an ID of `myGallery`, this reference string would be used: **`myGallery->specialImage`**
    - To reference the fifth image in a gallery with an ID of `myGallery`, this reference string would be used: **`myGallery->4`**

Linking Directly to a Specific Gallery or Image

Once you have determined the UniqueID string for a specific gallery or image, to reference that gallery or image you simply add a URL parameter named `xfs4_gid` to the page containing the slideshow. Suppose a slideshow is in `myslideshow.html`. To link to the `myGallery` gallery within that slideshow, you would use the following URL:

```
myslideshow.html?xfs4_gid=myGallery
```

If you wanted to link to the second image in that gallery, would modify the UniqueID string to be:

```
myslideshow.html?xfs4_gid=myGallery->1
```

Using these types of links allows you to create and share standard HTML links which reference specific galleries or images in a slideshow.

Notes for Developers and Advanced Users

Note for Developers: Behind the scenes, passing a `xfs4_gid` URL parameter makes the slideshow define a default `XMLFlashSlideshow_v4_getInitialImage` function (see the Slideshow / Javascript Interfacing topic for more information). Passing a URL parameter permalink only works if you have not already defined a `XMLFlashSlideshow_v4_getInitialImage` function; if a function already exists, it will not be overwritten.

Tip for Advanced Users: You can also use Javascript to control slideshows without having to reload the page. To learn more about this, see the Slideshow / Javascript Interfacing topic.

Slideshow / Javascript Interfacing

Important Note: This topic is only applicable to advanced users and developers. Experience with Javascript is required.

Tip: You can also reference specific images and galleries directly through the page URL by using Permalinks. See the Using Permalinks topic for more information. It is recommended that you review the permalinks topic for roundness of knowledge even if you don't intend to use the feature.

The XFS v4 includes built-in support for interaction with the host HTML page through javascript. This allows the HTML page to use javascript to both track the slideshow's status and control it. This feature is implemented through the `com.dwuser.ss4.core.SlideshowInterface` class which is detailed in the API Reference. This topic provides details on how to use this feature.

Call Types

There are two types of supported calls:

- **Outgoing calls** - These calls are made automatically from the slideshow to the host page to notify of the current state or request information. You define these methods within your page.
- **Incoming calls** - The slideshow automatically listens for calls to certain methods in the host page. When these methods are called by javascript in the host page, actions are automatically taken by the slideshow. You call these methods within your own scripts.

Important: In the code below, image and gallery **UniqueID**s are referenced. These are special identifiers which can be used to reference specific galleries or even specific images in a specific gallery. They are based on either the unique ID values assigned to the images and/or galleries, or their numeric indices within the slideshow or an individual gallery. Before proceeding, it is important to review *The UniqueID Format* in the Using Permalinks topic. This explains in detail the format used to reference individual galleries and images.

Details and Examples for Outgoing Calls

Note: Use of javascript interaction is not currently suported when there is more than one slideshow in a given page. To work around this, you can place each slideshow in a separate HTML page, when display them in the main page using <iframe>'s.

Method Name/Signature	Description

`XMLFlashSlideshow_v4_getInitialViewID():String`	As the slideshow is initializing, this method is called automatically in the host page. If the method is defined and returns a valid string value, the layout view corresponding to this return value will be opened as the initial view. For example, you might have defined the following view in your Layout XML in the XML configuration file: `<view id="demoView" ... >`. To dynamically make this view display first, the following method should be added to a javascript code block in your HTML page:
	`<script language="javascript">`
	`function XMLFlashSlideshow_v4_getInitialViewID()`
	`{`
	` return "demoView";`
	`}`
	`</script>`

`XMLFlashSlideshow_v4_getInitialImage():String`	Before the slideshow automatically opens the initial gallery, this method is called automatically in the host page. It allows you to override which gallery and/or image is displayed initially. If the method is defined and returns a valid UniqueID value, the gallery and/or image corresponding to the returned UniqueID will be opened. For example, the following javascript code block in your HTML page would make the third image in the first gallery open initially:

```
<script language="javascript">
function XMLFlashSlideshow_v4_getInitialImage()
{
 return "0->2";
}
</script>
```

This code, because it only specifies a gallery id, would open the first image in the gallery with an ID value of myGallery :

```
<script language="javascript">
function XMLFlashSlideshow_v4_getInitialImage()
{
 return "myGallery";
}
</script>
```

Note: If no custom XMLFlashSlideshow_v4_getInitialImage function has been defined and a xfs4_gid permalink URL parameter has been passed, a function is automatically defined to enable automatic permalinking. For more information, see Using Permalinks.

XMLFlashSlideshow_v4_notifyReadyForImageSet():void

This method is automatically called in the host page once the slideshow has initialized to the point of being ready to accept incoming calls. See the table below for more information on incoming calls, which allow your scripts to control the slideshow. For example, the following javascript code block could be placed in your HTML page:

```
<script language="javascript">
function XMLFlashSlideshow_v4_notifyReadyForImageSet()
{
 // This method doesn't return anything...
 alert('The slideshow says it is ready for incoming calls!');
}
</script>
```

XMLFlashSlideshow_v4_syncGallery(JSONInfoObject:String):void

This method is automatically called in the host page whenever a gallery is opened. It allows your page to track the current gallery that is open in the slideshow. You could use this to display pertinent gallery-related information in the HTML page.

The passed string is a JSON string which evaluates to an object with the following properties:

- uniqid - This is the standard UniqueID associated with the gallery. It is either the gallery's numeric index, or its associated unique ID value. This value can be used for other calls, such as the XMLFlashSlideshow_v4_requestGallery(uniqid:String) call documented below.
- uniqidOLD - This is the UniqueID for the gallery in the old **deprecated** format (e.g. gallery_0). It is only included for backwards compatibility.
- galleryID - Same as the uniqid property. Included for completeness.

To use the information in the JSON object, an eval() call must be used on the string. For example, the following javascript code block could be placed in your HTML page to display an alert every time a new gallery is opened:

```
<script language="javascript">
function XMLFlashSlideshow_v4_syncGallery(jsonString)
{
 // This method doesn't return anything...
 var infoObject = eval(jsonString);
 alert('The slideshow has just opened a gallery.  Here is its UniqueID: ' + infoObject.uniqid);
}
</script>
```

`XMLFlashSlideshow_v4_syncImagesStart(JSONInfoObject:String):void`

This method is automatically called in the host page whenever an image starts to be opened/synced. It allows your page to track the current image that is open in the slideshow. You could use this to display pertinent image-related information in the HTML page, or to track which image is open and store the state in a cookie, etc. Or, you could use a package such as SWFAddress to enable back/forward button browser navigation.

The passed string is a JSON string which evaluates to an object with the following properties:

- `uniqid` - This is the standard UniqueID associated with the image. It is either the gallery's numeric index, or its associated unique ID value, followed by `->` and the image's numeric index or its associated unique ID value. This value can be used for other calls, such as the `XMLFlashSlideshow_v4_setImage(uniqid:String)` call documented below.
- `uniqidOLD` - This is the UniqueID for the image in the old **deprecated** format (e.g. `img_0_5`). It is only included for backwards compatibility.
- `imageID` - This is the image's ID. It is the image's associated unique ID value (if one was defined), or otherwise its numeric index.
- `galleryID` - This is the gallery's ID. It is the gallery's associated unique ID value (if one was defined), or otherwise its numeric index.

To use the information in the JSON object, an `eval()` call must be used on the string. For example, the following javascript code block could be placed in your HTML page to display an alert every time a new image starts to open:

```
<script language="javascript">
function XMLFlashSlideshow_v4_syncImagesStart(jsonString)
{
 // This method doesn't return anything...
 var infoObject = eval(jsonString);
 alert('The slideshow has just started to open a new image.  Here is its UniqueID: ' + infoObject.uniqid);
}
</script>
```

`XMLFlashSlideshow_v4_syncImages(JSONInfoObject:String):void`

This method is automatically called in the host page whenever an image has been opened/synced. It is called shortly after the above `syncImagesStart` method. It allows your page to track the current image that is open in the slideshow. You could use this to display pertinent image-related information in the HTML page, or to track which image is open and store the state in a cookie, etc. Or, you could use a package such as SWFAddress to enable back/forward button browser navigation.

The passed string is a JSON string which evaluates to an object with the following properties:

- `uniqid` - This is the standard UniqueID associated with the image. It is either the gallery's numeric index, or its associated unique ID value, followed by `->` and the image's numeric index or its associated unique ID value. This value can be used for other calls, such as the `XMLFlashSlideshow_v4_setImage(uniqid:String)` call documented below.
- `uniqidOLD` - This is the UniqueID for the image in the old **deprecated** format (e.g. `img_0_5`). It is only included for backwards compatibility.
- `imageID` - This is the image's ID. It is the image's associated unique ID value (if one was defined), or otherwise its numeric index.
- `galleryID` - This is the gallery's ID. It is the gallery's associated unique ID value (if one was defined), or otherwise its numeric index.

To use the information in the JSON object, an `eval()` call must be used on the string. For example, the following javascript code block could be placed in your HTML page to display an alert every time a new image has opened:

```
<script language="javascript">
function XMLFlashSlideshow_v4_syncImages(jsonString)
{
 // This method doesn't return anything...
 var infoObject = eval(jsonString);
 alert('The slideshow has just opened a new image.  Here is its UniqueID: ' + infoObject.uniqid);
}
```

	`</script>`

`XMLFlashSlideshow_v4_viewReady(currentViewID:String):void`	This method is automatically called in the host page whenever a new layout view is opened/displayed. It allows your page to track which layout view is currently open. You could use this in conjunction with the above `getInitialViewID()` or below `setLayoutView` method to restore a slideshow's layout state when a user returns to a page.
	The passed string parameter is the `id` value of the new view.
	For example, the following javascript code block could be placed in your HTML page to display an alert every time a new layout view is opened:
	`<script language="javascript">`
	`function XMLFlashSlideshow_v4_viewReady(curentViewID)`
	`{`
	` // This method doesn't return anything...`
	` alert('The slideshow has just opened a new layout view.  Here is its ID value: ' + currentViewID);`
	`}`
	`</script>`

Details and Examples for Incoming Calls

Method Name/Signature	Description

`XMLFlashSlideshow_v4_requestGallery(uniqid:String = "0"):void`	This call cannot be made until the `XMLFlashSlideshow_v4_notifyReadyForImageSet()` outgoing call has been made (see above). Calling this method opens the specific gallery identified by the passed gallery UniqueID. The unique ID can be the gallery's linkage ID, or its numeric index. For example, the following snippet (which would need to be placed within some other block of code) would open the second gallery:
	`<script language="javascript">`
	`...`
	`// Open the second gallery`
	`XMLFlashSlideshow_v4_requestGallery("1");`
	`...`
	`</script>`
	If you wanted to open a gallery with an ID value of `myGallery`, you could use the following code:
	`<script language="javascript">`
	`...`
	`// Open the second gallery`
	`XMLFlashSlideshow_v4_requestGallery("myGallery");`
	`...`
	`</script>`
	*Note: The deprecated format which was previously used is `gallery_`**N** where **N** is the numeric index of the gallery. While this format still works, it is preferable to use the new format of either a simple numeric index or a gallery ID string.*

`XMLFlashSlideshow_v4_setImage(uniqid:String = "0->5"):void`	This call cannot be made until the `XMLFlashSlideshow_v4_notifyReadyForImageSet()` outgoing call has been made (see above). Calling this method opens the specific image identified by the passed image UniqueID. For example, the following snippet (which would need to be placed within some other block of

code) would open the third image in the second gallery:

```
<script language="javascript">
...
// Open the third image in the second gallery
XMLFlashSlideshow_v4_setImage("1->2");
...
</script>
```

If you wanted to open an image with an ID value of `myImage` within a gallery with an ID value of `myGallery`, you could use the following code:

```
<script language="javascript">
...
// Open the second gallery
XMLFlashSlideshow_v4_setImage("myGallery->myImage");
...
</script>
```

*Note: The deprecated format which was previously used is img_**N**_**I** where **N** is the numeric index of the gallery and **I** is the numeric index of the image within the gallery. While this format still works, it is preferable to use the new format of a gallery index or ID followed by **->** and an image index or ID.*

`XMLFlashSlideshow_v4_setLayoutView(viewID:String = "someViewID"):void`

Calling this method opens the layout view associated with the passed view ID. This method is only to be used for *changing* the current layout view of an already-initialized slideshow. If you wish to explicitly specify the *initial* layout view, use the above-described `getInitialViewID` method. For example, the following snippet (which would need to be placed within some other block of code) would open the layout view with an ID of `specialLayoutView` :

```
<script language="javascript">
...
// Open the layout view with an ID of 'specialLayoutView'
XMLFlashSlideshow_v4_setLayoutView("specialLayoutView");
...
</script>
```

`XMLFlashSlideshow_v4_getXML():String`

This call cannot safely be made until the `XMLFlashSlideshow_v4_notifyReadyForImageSet()` outgoing call has been made (see above). Calling this method returns the full XML document being used for the current slideshow as a string. This XML document incorporates any external data sources that have been loaded and transformed into the slideshow XML format. (For more information about the slideshow XML format, see the Developer Reference.) This method is useful, for example, in creating a dynamic HTML gallery selector: the XML can be requested and then parsed with a tool such as jQuery to find the galleries, then this list used to build a gallery selector that makes `XMLFlashSlideshow_v4_requestGallery(...)` calls. The method is also useful for debugging. For example, the following snippet (which would need to be placed within some other block of code) would display the entire XML document in an alert box. Note that this may be very large, and to clear it off your screen you may need to press the enter/return key.

```
<script language="javascript">
...
// Display the slideshow's full XML in an alert box; press enter to clear the box if you can't see the 'dismiss' button.
alert(XMLFlashSlideshow_v4_getXML());
...
```

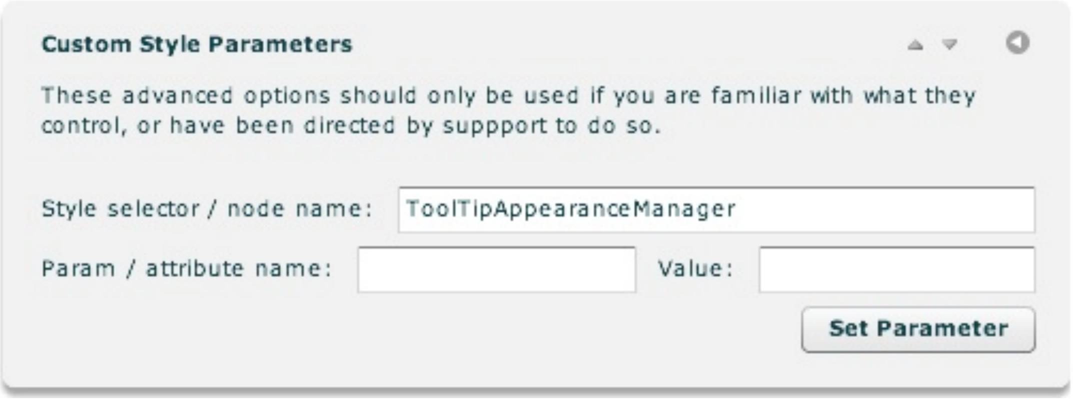
	`</script>`

XMLFlashSlideshow_v4_setPlaying(playing:Boolean):void	Calling this method enables or disables autoplay for a slideshow. For example, the following snippet (which would need to be placed within some other block of code) would enable autoplay:
	`<script language="javascript">`
	`...`
	`// Enable autoplay for the slideshow (start it playing)`
	`XMLFlashSlideshow_v4_setPlaying(true);`
	`...`
	`</script>`

Customizing Slideshow Tooltips

There are two main style selectors which are used to control the customize the tooltips shown throughout a slideshow. These are the `HTMLToolTip` selector (which controls the appearance of the tooltips) and the `ToolTipAppearanceManager` selector (which controls the behavior of the tooltips). For developers and advanced users, these two selectors are covered in detail in the `com.dwuser.ss4.ui.extras.HTMLToolTip` and `com.dwuser.ss4.managers.ToolTipAppearanceManager` classes respectively in the API Reference.

These options can be customized in the wizard using the Custom Style Parameters pod in the main Global Configuration and Playback Settings section:



Custom Style Parameters

These advanced options should only be used if you are familiar with what they control, or have been directed by suppport to do so.

Style selector / node name: ToolTipAppearanceManager

Param / attribute name:

Value:

In the Style Selector box, you will enter either `HTMLToolTip` or `ToolTipAppearanceManager` . The following settings are most commonly used for the **`HTMLToolTip`** style selector:

Property	Type	Description
`backgroundColor`	Hexadecimal color, e.g. `#FF0000`	Specifies the background color of all tooltips.
`color`	Hexadecimal color, e.g. `#000000`	Specifies the color of the text in all tooltips.
`fontFamily`	Font name, e.g. `Verdana`	Specifies the font used for the text in all tooltips.
`fontSize`	Numeric value	Specifies, in pixels, the size of the font used for the text in all tooltips.

The following settings are most commonly used for the **ToolTipAppearanceManager** style selector:

Property	Type	Description
showDelay	Numeric value, in milliseconds	Specifies the delay, in milliseconds, before tooltips appear. To make the tooltips appear immediately, set the value to 0. The default value is 500.
hideDelay	Numeric value, in milliseconds	Specifies how long tooltips will appear before automatically hiding. To make the tooltips appear until the user moves their mouse off of the object, set the parameter to a very large value. The default value is 10000.
trackMouse	Boolean - true or false	Specifies if tooltips should track the mouse after appearing, or remain in the place where they initially appear. The default value is false.

As mentioned earlier, *all* available settings for both of these selectors are listed in the API Reference.

Purchase Buttons / E-Commerce Integration

This topic demonstrates integrating a PayPal purchase button with a slideshow, allowing for the sale of prints, products, etc. The example uses a PayPal purchase button, but the techniques are applicable to any type of e-commerce system.

The basic principles are as follows:

- A custom `itemName` and `itemNumber` and `itemPrice` field is added to each `<img>` node in the XML file. This associates an item name and item number with each image / slide.
- A standard PayPal purchase button form is inserted in the HTML page then wrapped in HTML code to hide it, making it invisible to the user.
- A Javascript function is created. This function takes three parameters -- an item number and name, and the item price. It dynamically updates the PayPal purchase button to reflect this new information, then submits the form (redirecting the user to PayPal to complete the purchase).
- A DynamicLinkButton element is added to the slideshow. When clicked, this button reads the current slide's `itemName` and `itemNumber` and `itemPrice` and calls the javascript function to simulate a click on the PayPal purchase button.

Adding The Custom Fields

First, open the XML configuration file for editing. This can be done by selecting the slideshow in Design View and clicking the 'Edit Manually...' button in the Properties panel. Near the bottom of the file, locate the `<galleries>` section and find the `<img ... />` lines in this section. On each of these nodes, add a new `itemName`, `itemNumber` and `itemPrice` field. For example, the bolded code might be added:

```
...
<img src="toycar_red.jpg" title="A title for this slide... tell about the car" itemName="Red Toy Car"
itemNumber="TOYCAR1" itemPrice="15.00" />
<img src="toycar_blue.jpg" title="A title for this slide... tell about the car" itemName="Blue Toy Car"
itemNumber="TOYCAR2" itemPrice="22.50" />
...
```

Save and close the XML configuration file.

Adding Purchase Button Code and Javascript Purchase Function

Next, you'll need to add the standard PayPal purchase button code to your page (within the `<body>` tag). Some modifications have been made, allowing for an item number to be passed and making the form invisible. The bolded email address must be replaced with your PayPal email address. Additionally, the default submit button has been removed:

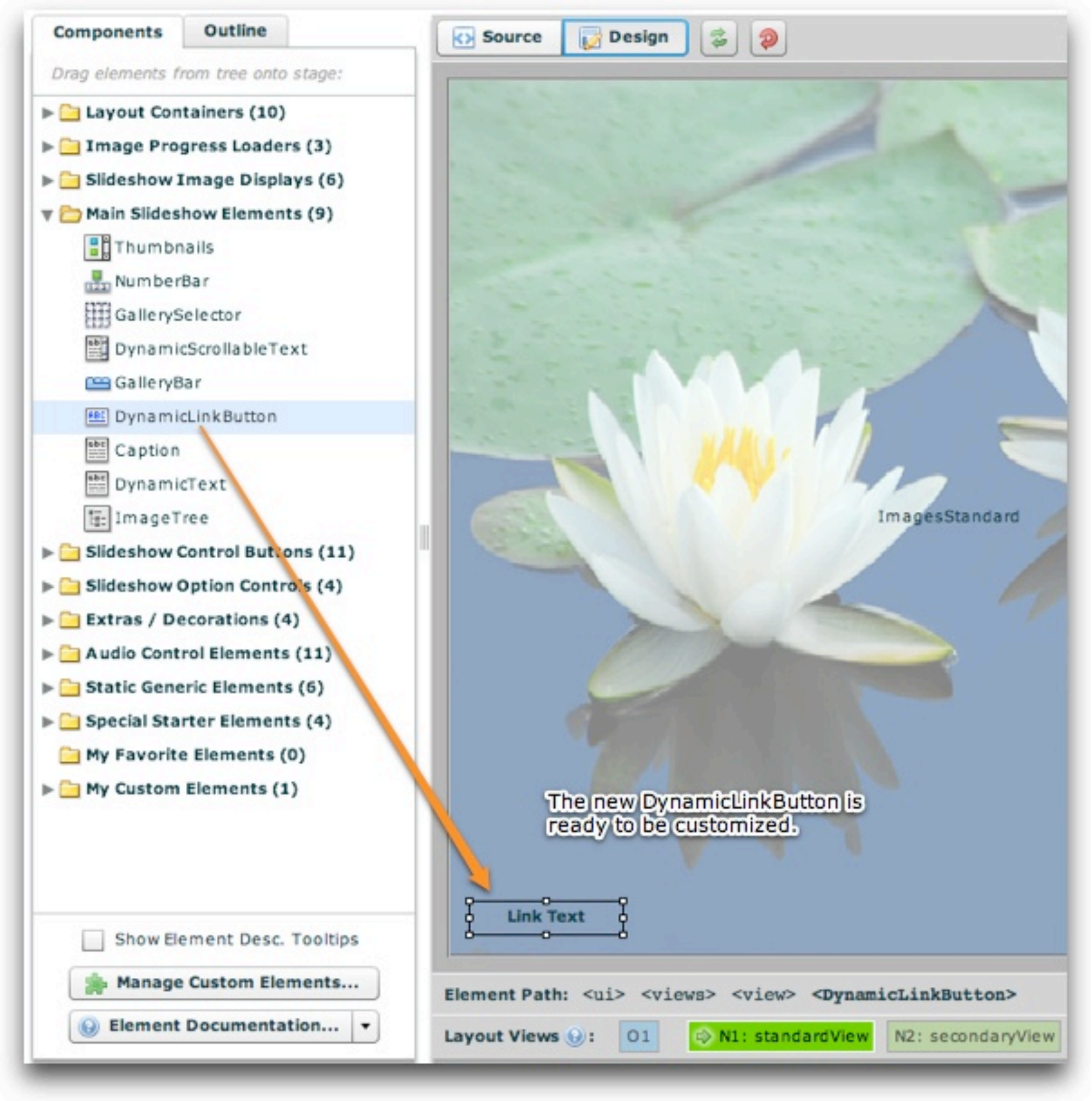
```
<div style="display: none;">
    <form name="_xclick" action="https://www.paypal.com/cgi-bin/webscr" method="post">
        <input type="hidden" name="cmd" value="_xclick">
        <input type="hidden" name="business" value="me@mybusiness.com">
        <input type="hidden" name="currency_code" value="USD">
        <input type="hidden" name="item_name" value="FillerName">
        <input type="hidden" name="item_number" value="FillerNumber">
        <input type="hidden" name="amount" value="12.99">
    </form>
</div>
```

The next step is to create the javascript function. In the `<head>` of your document, add the following code. JQuery is being used to simplify the code.

```
<script language="javascript" type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min.js"></script>
<script language="javascript" type="text/javascript">
function doPurchase(itemNumber, itemName, itemPrice)
{
     // Update the form with the new information
     $('input[name="item_number"]').val(itemNumber);
     $('input[name="item_name"]').val(itemName);
     $('input[name="amount"]').val(itemPrice);

     // Submit the form, redirecting the user to PayPal to complete the purchase
     $('form[name="_xclick"]').submit();

     // Return false so that the caller doesn't try to do any redirect
     return false;
}
</script>
```

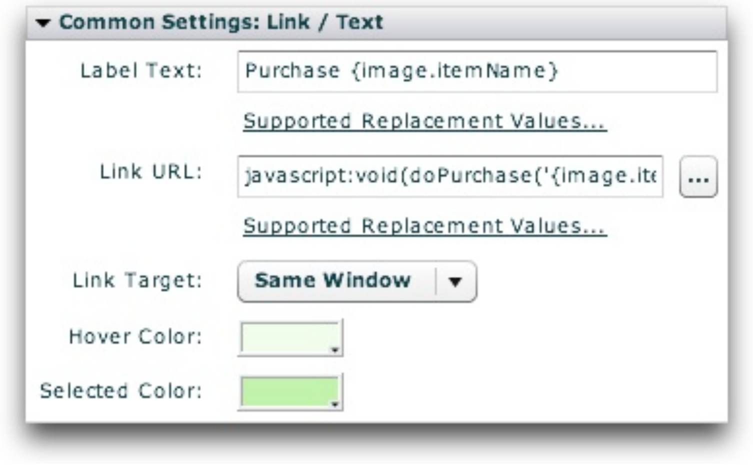
Creating The Purchase Button

Now, open the wizard and switch to the Layout / Presentation section. Select the layout view where you want to add the purchase button; this will usually be the same view as that which displays the images. Drag a new DynamicLinkButton element onto the stage from the Main Slideshow Elements section of the Components Panel:


Components
Outline
Drag elements from tree onto stage:
Layout Containers (10)
Image Progress Loaders (3)
Slideshow Image Displays (6)
Main Slideshow Elements (9)
Thumbnails
NumberBar
GallerySelector
DynamicScrollableText
GalleryBar
DynamicLinkButton
Caption
DynamicText
ImageTree
Slideshow Control Buttons (11)
Slideshow Option Controls (4)
Extras / Decorations (4)
Audio Control Elements (11)
Static Generic Elements (6)
Special Starter Elements (4)
My Favorite Elements (0)
My Custom Elements (1)
Show Element Desc. Tooltips
Manage Custom Elements...
Element Documentation...
Source
Design
ImagesStandard
Link Text
Element Path: <ui> <views> <view> <DynamicLinkButton>
Layout Views
O1
N1: standardView
N2: secondaryView

The new DynamicLinkButton is ready to be customized.

Next, the button's width is increased so there's more space for the label. Then, the Label Text and Link URL properties are set in the Properties Panel:



▼ Common Settings: Link / Text

Label Text: Purchase {image.itemName}

Supported Replacement Values...

Link URL: javascript:void(doPurchase('{image.ite …

Supported Replacement Values...

Link Target: Same Window ▼

Hover Color:

Selected Color: 🟩

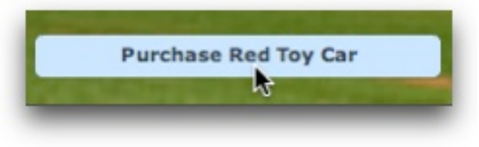
The Label Text is set to be `Purchase {image.itemName}`. By using the replacement value, this will make the button display for example 'Purchase Red Toy Car'. The Link URL is set to:

```
javascript:void(doPurchase('{image.itemNumber}', '{image.itemName}', '{image.itemPrice}'));
```

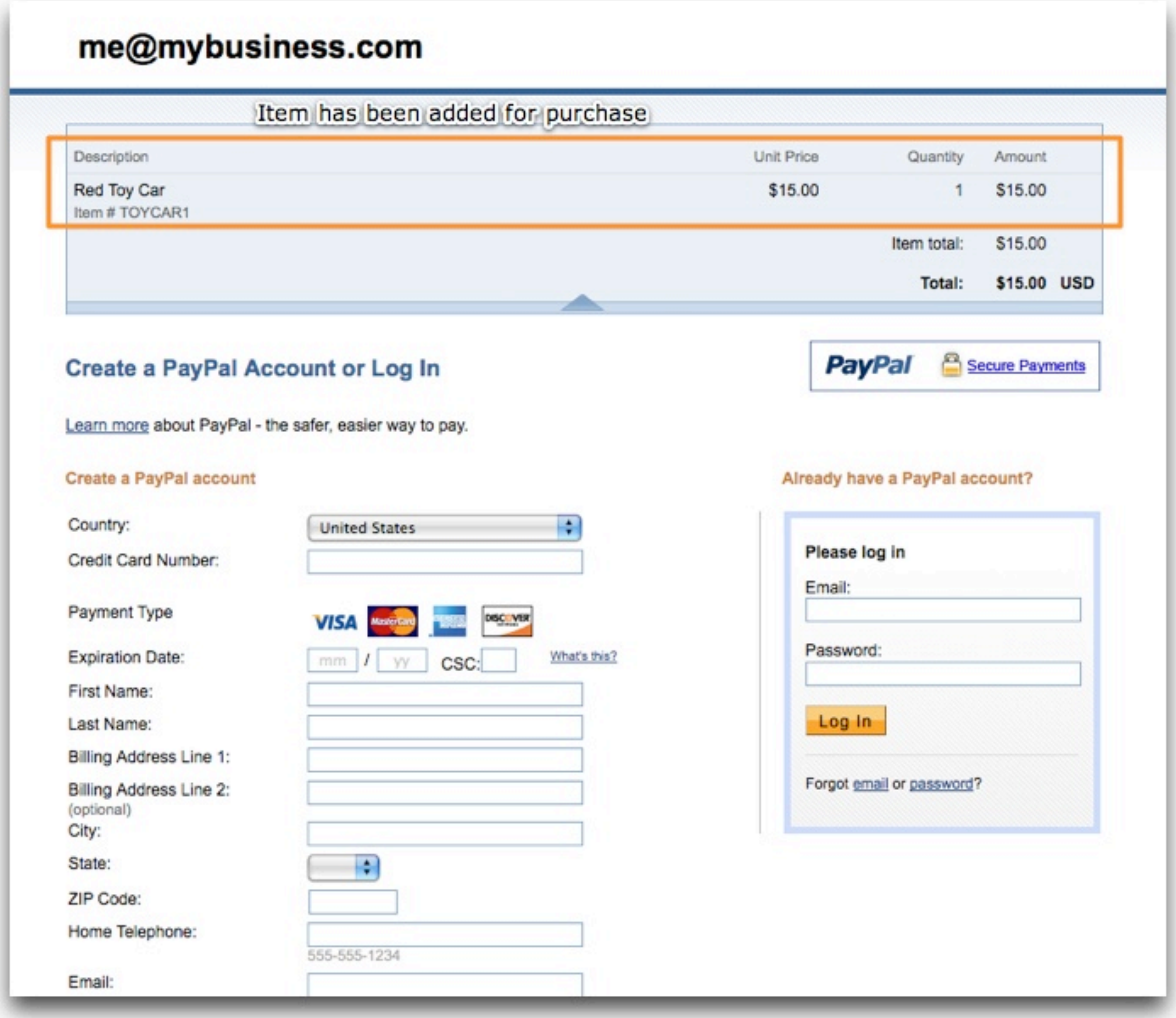
This calls the javascript function earlier defined in the page. Again, the replacement values are used to pass the special item information attributes to the purchase function. Note that quotes are still used around each value; failure to do so will result in errors.

The Finished Product

When previewed in a browser, the purchase button appears like this:



When clicked, you will be taken to PayPal where the purchase can be completed. All of the item's information has automatically been specified:



me@mybusiness.com

Item has been added for purchase

Description	Unit Price	Quantity	Amount
Red Toy Car Item # TOYCAR1	$15.00	1	$15.00
		Item total:	$15.00
		Total:	**$15.00 USD**

Create a PayPal Account or Log In

Learn more about PayPal - the safer, easier way to pay.

Create a PayPal account

Country: United States

Credit Card Number:

Payment Type

Expiration Date: mm / yy CSC: What's this?

First Name:

Last Name:

Billing Address Line 1:

Billing Address Line 2: (optional)

City:

State:

ZIP Code:

Home Telephone: 555-555-1234

Email:

Already have a PayPal account?

Please log in

Email:

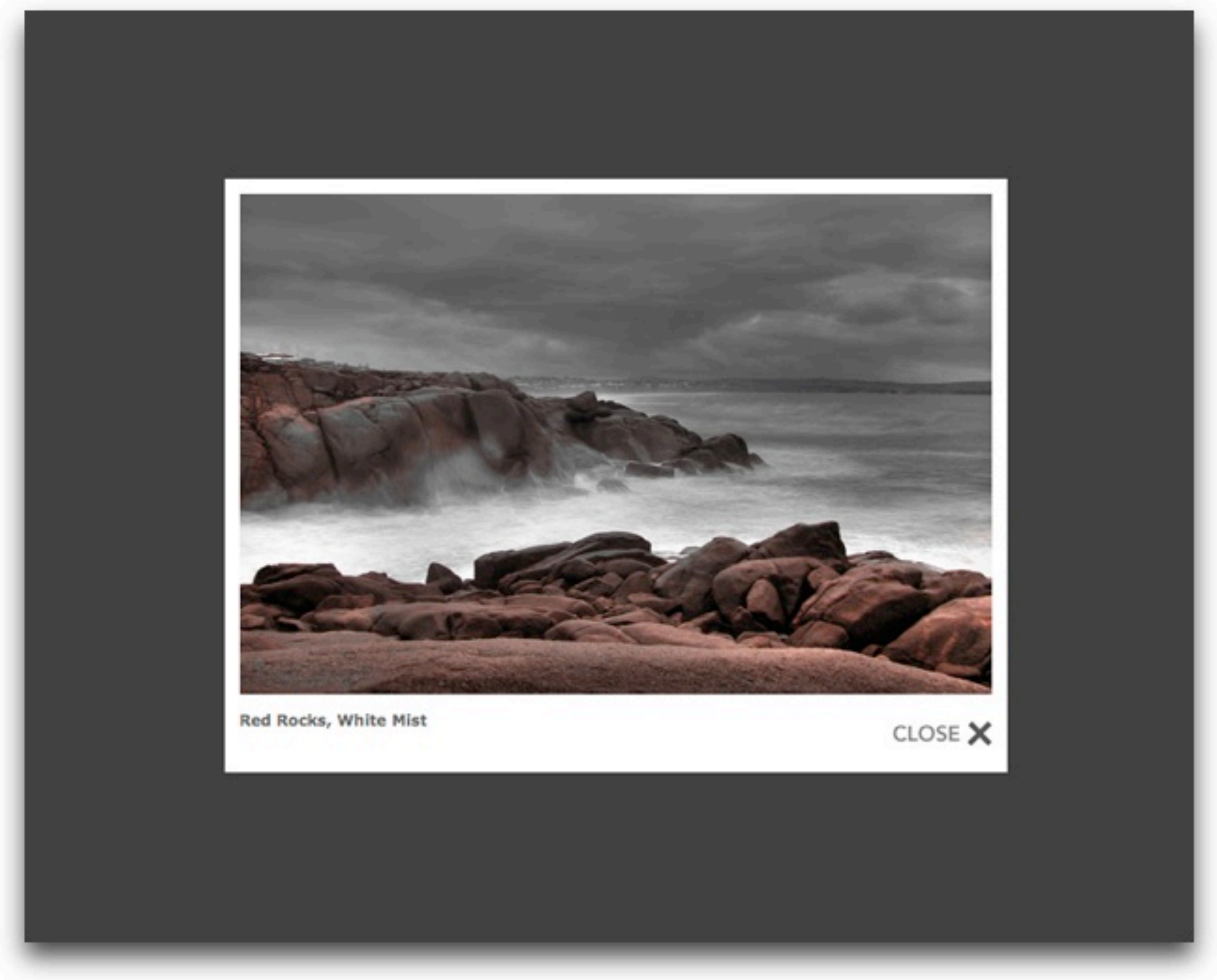
Password:

Forgot email or password?

Lightbox Integration

If you have a small slideshow in your page, you may wish to use a Lightbox-style popup when users click on an image, allowing them to see a larger view of the photo. For simplified integration and reduced file size, it is highly recommended that you use a Lightbox2 clone for jQuery called Slimbox2. (Slimbox greatly simplifies integration by having a direct javascript API, whereas most other scripts are only designed for HTML integration through setting a `<a ... rel="...">` value, which cannot be used inside a slideshow.)

In our example, we have a small slideshow in the example page (it is only 300 x 200 px). When completed, clicking on an image in the slideshow will result in a popup like this:



Installing Slimbox Support into the HTML Page

First, Slimbox must be installed to the HTML page which holds the slideshow. Full instructions are available on the Slimbox page, but they have been summarized here:

- Download Slimbox from http://code.google.com/p/slimbox/downloads/detail?name=slimbox-2.04.zip . Make sure that you download version 2 (for jQuery), **not** 1.7.
- In the Slimbox ZIP package, locate the `js/slimbox2.js` file and copy it to your site.
- In the Slimbox ZIP package, locate the `css/` folder and copy it to your site (including all files in the folder).
- If applicable, upload the files you just added.
- In your HTML page, add the following code to the `<head>` of the page. The bolded values need to be updated to correctly point to the `slimbox2.js` and `slimbox2.css` files you just added to your site; the first line includes the required jQuery framework:

  ```
  <script language="javascript" type="text/javascript"
  src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min.js"></script>
  <script language="javascript" type="text/javascript" src="js/slimbox2.js"></script>
  <link rel="stylesheet" href="css/slimbox2.css" type="text/css" media="screen" />
  ```

Note: See the Slimbox usage guide for more information about customizing the appearance and behavior of the lightbox popups.

Launching a Slimbox / Lightbox Window from the Slideshow

In the Image Properties dialog for the image/slide you wish to use, select the Image Link tab:

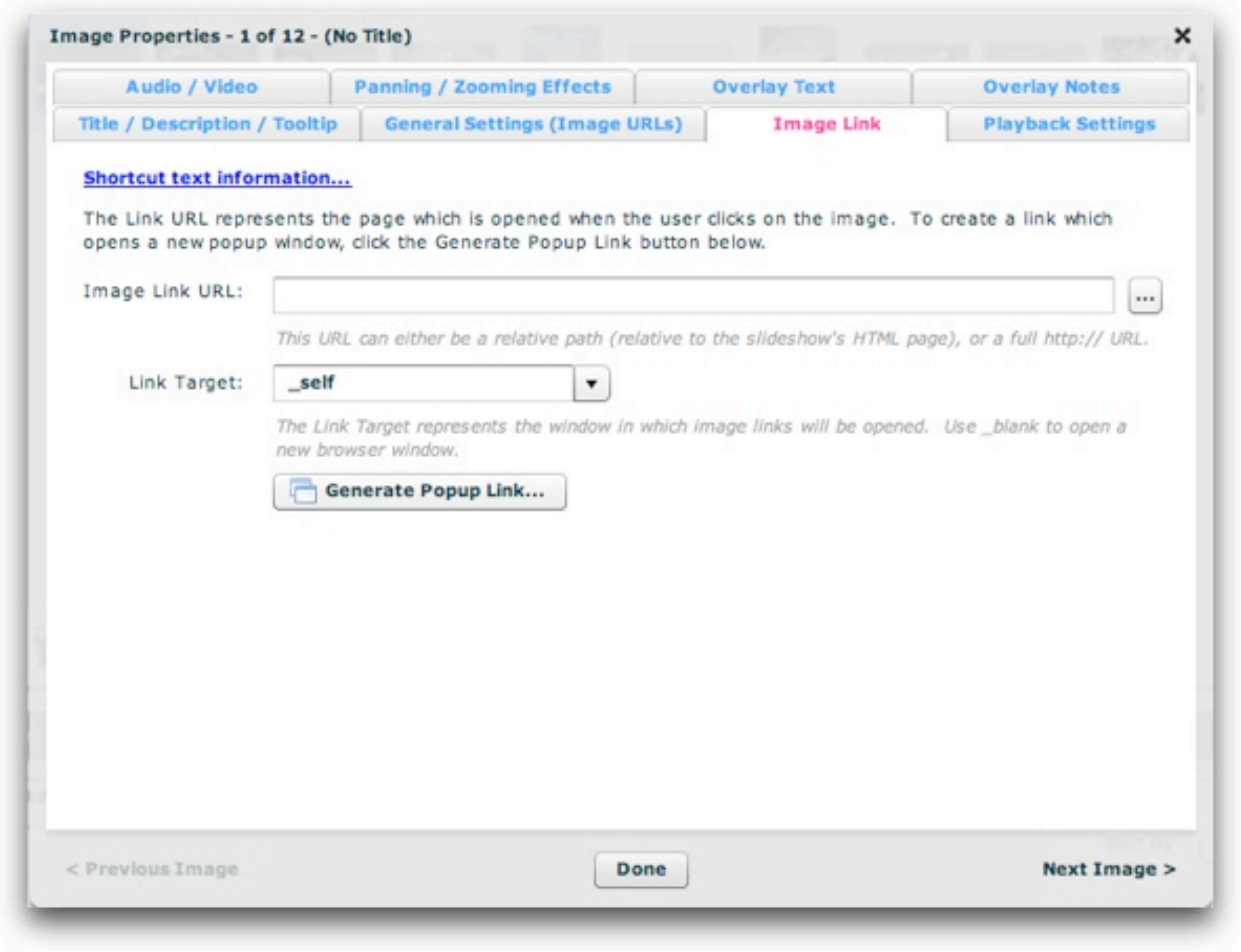


Image Properties - 1 of 12 - (No Title)
Audio / Video
Panning / Zooming Effects
Overlay Text
Overlay Notes
Title / Description / Tooltip
General Settings (Image URLs)
Image Link
Playback Settings
Shortcut text information...
...
Generate Popup Link...
< Previous Image
Done
Next Image >

The Link URL represents the page which is opened when the user clicks on the image. To create a link which opens a new popup window, click the Generate Popup Link button below.

Image Link URL:

This URL can either be a relative path (relative to the slideshow's HTML page), or a full http:// URL.

Link Target: _self

The Link Target represents the window in which image links will be opened. Use _blank to open a new browser window.

In the Image Link field, enter the following:

```
javascript:void(jQuery.slimbox('{image.srcFull}', '{image.title}'));
```

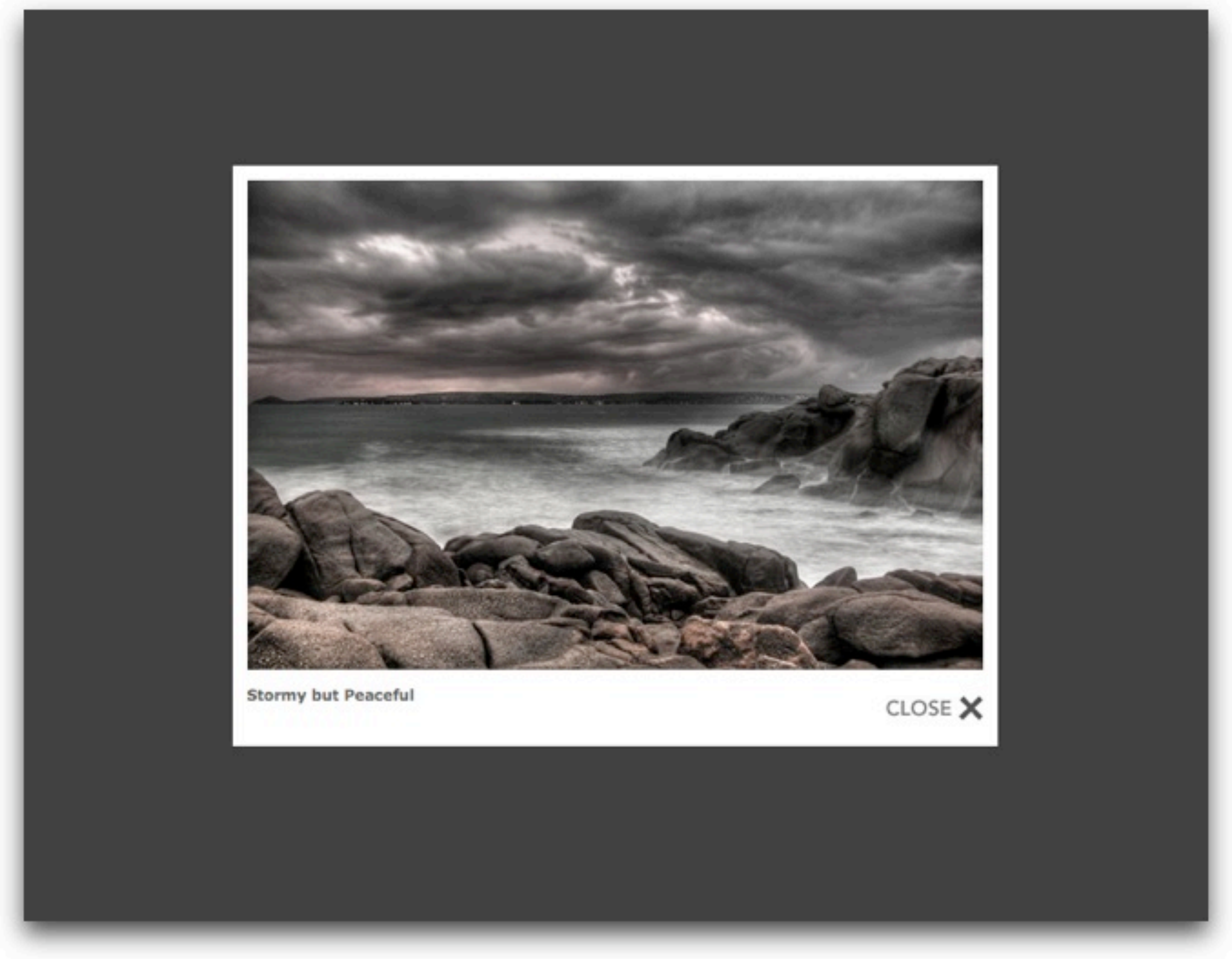
This will open a lightbox popup which displays the image and its title. If you only want the image (without the title), use the following:

```
javascript:void(jQuery.slimbox('{image.srcFull}'));
```

Note that in both cases the image information is dynamically referenced using Replacement Values, and quotes are used around the string values.

Also make sure that Link Target is set to `_self` then save and exit.

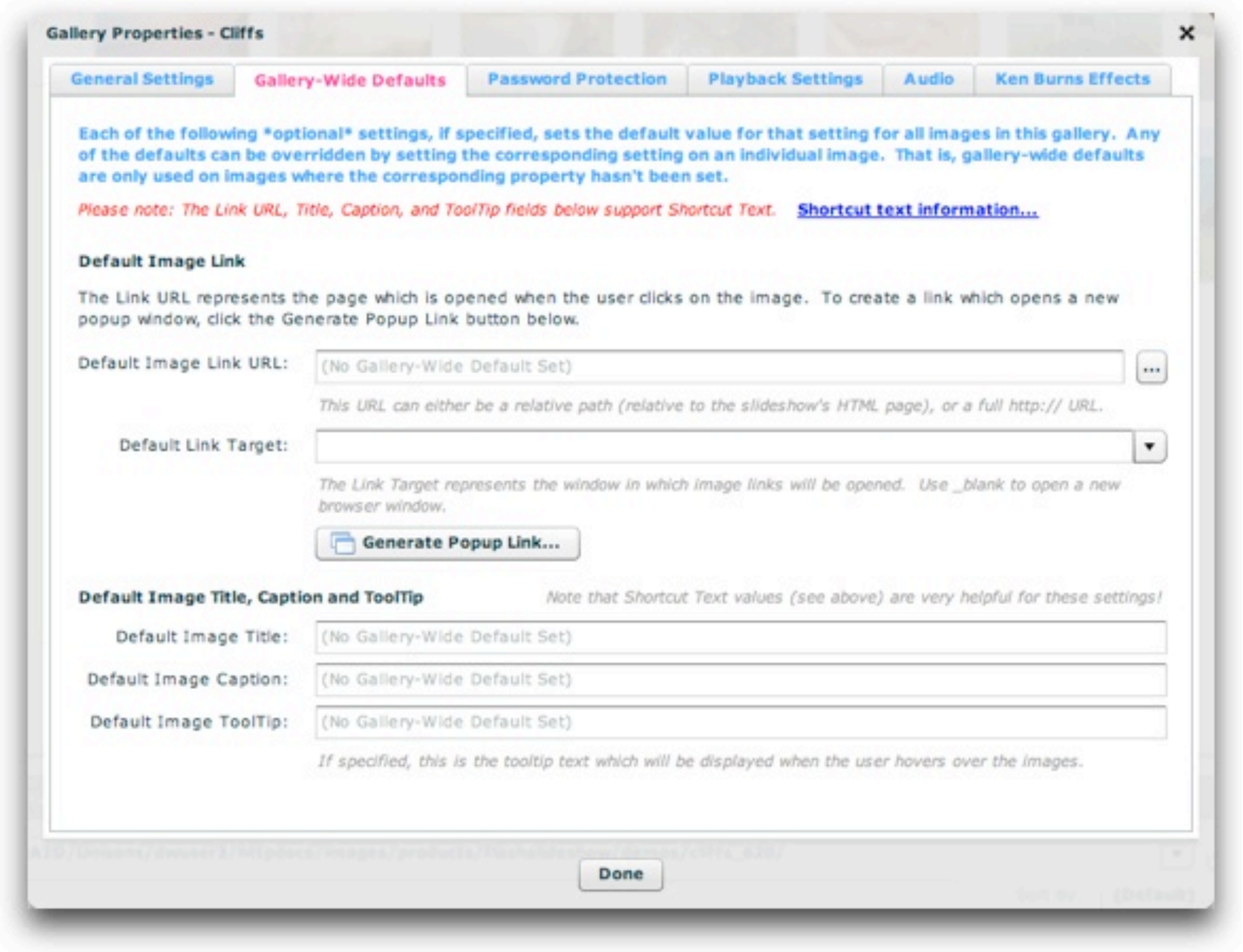
When you view the slideshow in a browser, clicking on an image will open a lightbox popup and pause the slideshow:



To close the popup, click the darkened area outside the popup, click the Close button or use the Esc key. If you wish for the user to be able to resume the slideshow, you must include a PlayPauseButton element. Also note that all images are displayed in the lightbox in isolation (e.g. there are no next/previous buttons). The lightbox integration is designed to enlarge a single photo, not become an alternate slideshow.

Using the Lightbox Effect for All Photos in a Gallery

To apply the lightbox effect to all photos in a gallery, open the gallery you wish to apply the effect to and click the *More Gallery Settings...* button to launch the Gallery Properties dialog. Select the Gallery-Wide Defaults tab:



Each of the following *optional* settings, if specified, sets the default value for that setting for all images in this gallery. Any of the defaults can be overridden by setting the corresponding setting on an individual image. That is, gallery-wide defaults are only used on images where the corresponding property hasn't been set.

Please note: The Link URL, Title, Caption, and ToolTip fields below support Shortcut Text.

Default Image Link

The Link URL represents the page which is opened when the user clicks on the image. To create a link which opens a new popup window, click the Generate Popup Link button below.

Default Image Link URL: (No Gallery-Wide Default Set)

This URL can either be a relative path (relative to the slideshow's HTML page), or a full http:// URL.

Default Link Target:

The Link Target represents the window in which image links will be opened. Use _blank to open a new browser window.

Default Image Title, Caption and ToolTip *Note that Shortcut Text values (see above) are very helpful for these settings!*

Default Image Title: (No Gallery-Wide Default Set)

Default Image Caption: (No Gallery-Wide Default Set)

Default Image ToolTip: (No Gallery-Wide Default Set)

If specified, this is the tooltip text which will be displayed when the user hovers over the images.

Enter the Link URL and Link Target values as mentioned above in the corresponding *Default Image Link URL* and *Default Link Target* fields. The lightbox effect will now automatically appear for all photos in the gallery.

Integration with Existing Flash Projects (Pro only)

The Flash integration feature allow you to use XFS v4 slideshows as a part of existing Flash projects. This feature is only available in the Pro version of the software.

Note: If you want to integrate in a Flex project, see this topic.

Prerequisites

There are several prerequisites for Flash integration:

- As mentioned above, you must be using the Pro version of the XFS v4 software.
- The Adobe Flash CS3 or CS4 (or higher) authoring tool is required. Prior versions, including MX, MX 2004, and 8 are *not* supported because they don't support Actionscript 3.
- Your project must be an Actionscript 3 project. Actionscript 2 projects are not supported; this is a Flash player limitation.
- It is highly recommended that you have experience and some comfortability with Flash usage and basic Actionscript 3.0 script editing. Without this basic knowledge, it will be very difficult to create a working implementation. If you're not sure of your skills, view the Flash help or consider instead a standard slideshow insertion in Dreamweaver.

Creating the Slideshow / File Structure

You must use Dreamweaver to create the slideshow which will be integrated into the Flash project. First, create a separate HTML page to hold the slideshow. This page won't be viewed online; it will only be used to hold the slideshow, allowing you to make edits. **Important:** To ensure that all paths are correct and functional, it is **critical** that you save this file in the same directory as the HTML page which displays your Flash-created SWF. For example, if your main Flash file is viewed at http://example.org/index.htm, then the slideshow holder page should be something like http://example.org/slideshow_holder.htm. If your FLA project is located in some other folder than the display folder (e.g. http://example.org/flash/myProject.fla in the above example), then the slideshow will not load while working/previewing in the Flash development tool. Rest assured, though, that it will appear properly when viewed in the context of the deployment file structure.

When you deploy your Flash project, you will have to upload all of the slideshow-related files as detailed in the slideshow insertion portion of the manual.

Failing to upload all required files will result in the slideshow not appearing.

Once the slideshow has been inserted, select it and switch to Code View in Dreamweaver. Locate the initialization object where the slideshow HTML parameters are passed. This is the line which appears similar to the following:

```
XMLFlashSlideshow_v4({width:'300', height:'200', level:'a', xml:'v4flashslideshow/slideshow_data1.xml',
backgroundColor:'#000000', backgroundAlpha:'1', preloaderTextColor:'#999999'});
```

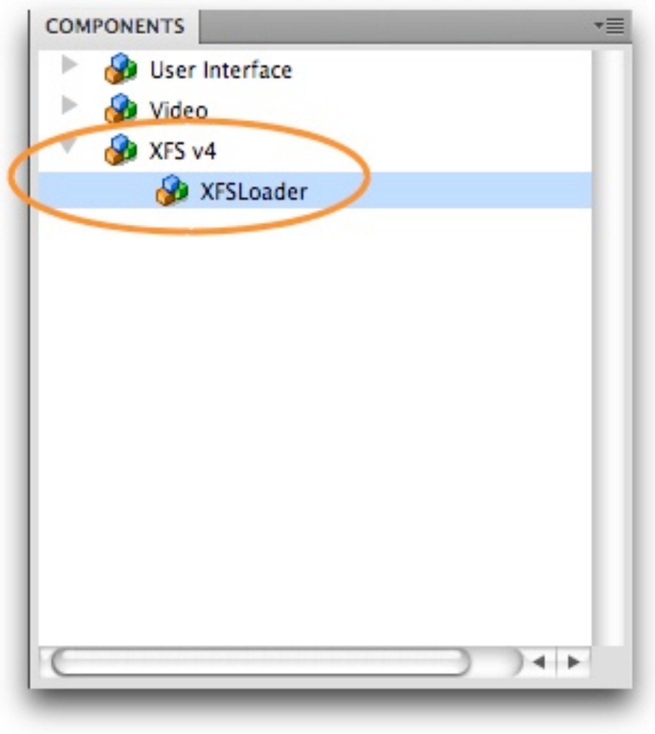
Note the bolded code, which starts with `{` and ends with `}` (the curly braces). This is the actual initialization object which passes important information such as the dimensions, the XML configuration file path, background color, etc. This is the slideshow parameter object which you will use later. When you make changes to the slideshow settings in the wizard, this parameter object may change, requiring you to update it in your Flash project (see below).

Configuring the Flash Project

First, you need to download the `XFSLoader` SWC file which enables a Flash project to load and display slideshows. The `XFSLoader.swc` file is available here. Download it and place it in a new `XFS v4/` folder in your Flash configuration's `Components/` directory. The path will be similar to the following:

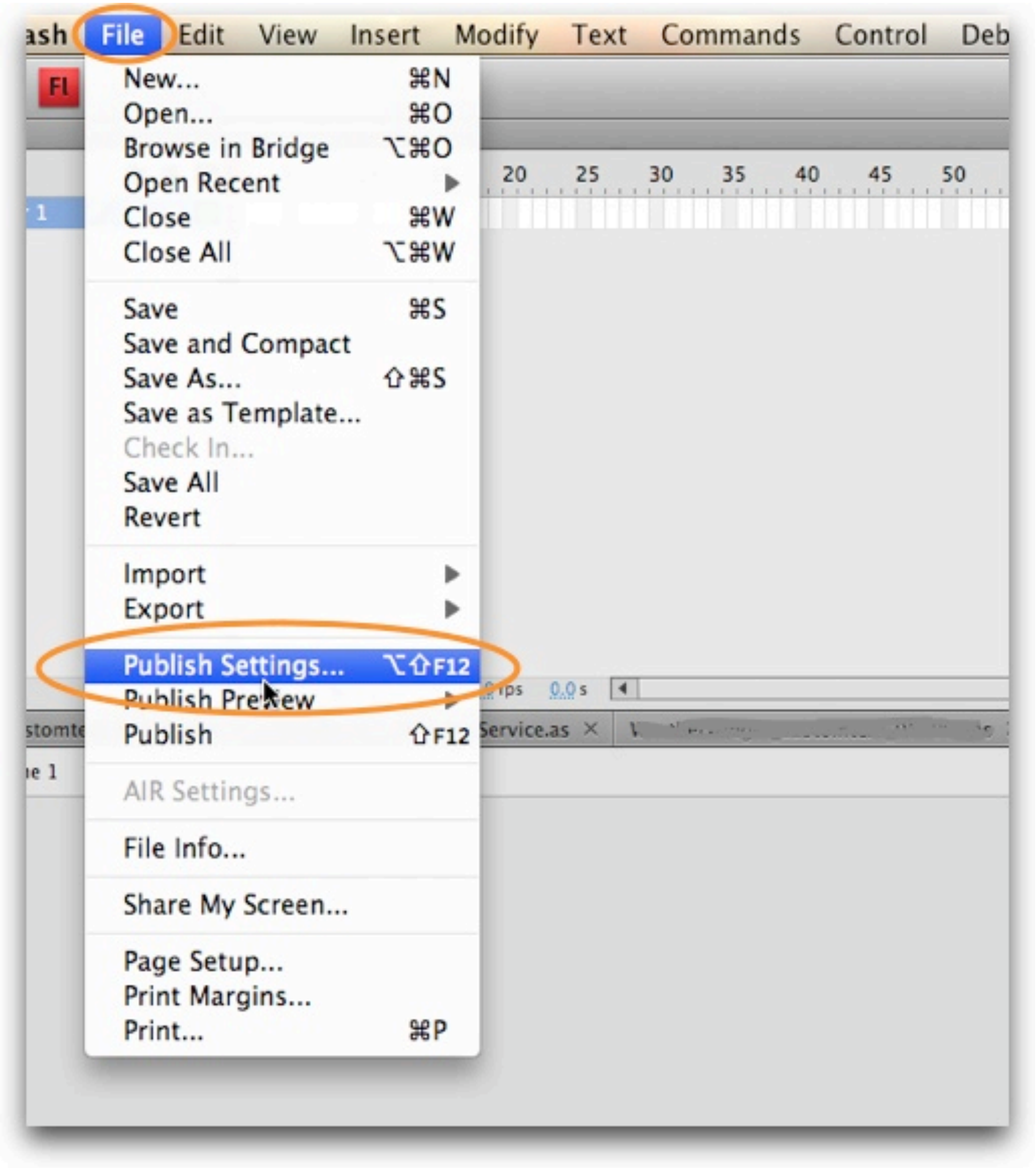
- **Windows XP:** C:/Documents and Settings/<username>/Local Settings/Application Data/Adobe/Flash CS*N*/<*language*>/Configuration/Components/XFS v4/
- **Windows Vista and 7:** C:/Users/<*username*>/Local Settings/Application Data/Adobe/Flash CS*N*/<*language*>/Configuration/Components/XFS v4/
- **Mac OSX:** /Users/<username>/Library/Application Support/Adobe/Flash CS*N*/<*language*>/Configuration/Components/XFS v4/

Restart Flash and open your FLA project. In the Components panel, locate the XFSLoader item in the new XFS v4 folder:

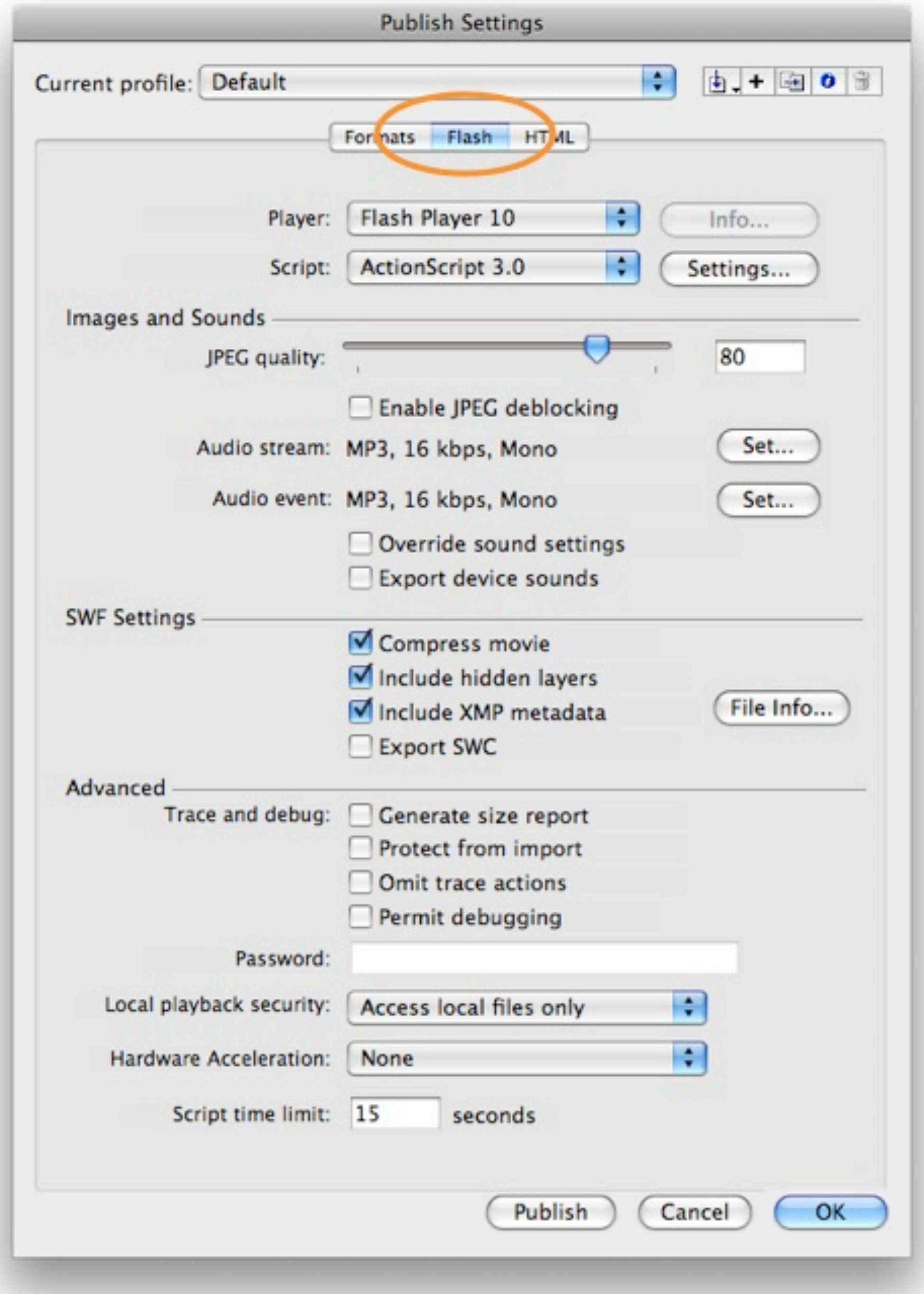


Open the Library and drag the XFSLoader component directly into the library. **Important:** Do **not** drag the XFSLoader component onto the stage. The component has no visual implementation; it is only used in code.

Next, you need to set the Flash Player version. To do this, open the Publish Settings dialog (accessed from the File menu):



The Publish Settings dialog will open; make sure that the Flash section is selected at the top:



Current profile: Default

Player: Flash Player 10

Script: ActionScript 3.0

Images and Sounds

JPEG quality: 80

☐ Enable JPEG deblocking

Audio stream: MP3, 16 kbps, Mono

Audio event: MP3, 16 kbps, Mono

☐ Override sound settings

☐ Export device sounds

SWF Settings

☑ Compress movie

☑ Include hidden layers

☑ Include XMP metadata

☐ Export SWC

Advanced

Trace and debug: ☐ Generate size report

☐ Protect from import

☐ Omit trace actions

☐ Permit debugging

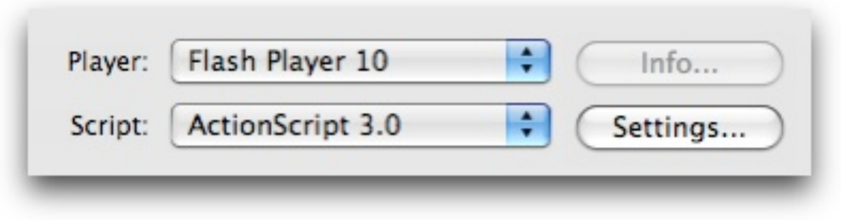
Password:

Local playback security: Access local files only

Hardware Acceleration: None

Script time limit: 15 seconds

The relevant settings are located at the top of the dialog:


Player: Flash Player 10

Script: ActionScript 3.0

Double-check that Script is set to Actionscript 3.0. The Player setting is dependent on the SWF version you specified in the 'Slideshow Type - Pro or Basic' pod in the Global Configuration and Playback Settings section of the main wizard when creating the slideshow. If you chose the Basic SWF, you can use the Flash Player 9 or higher. If you chose the Professional SWF, you **must** use the Flash Player 10 or higher. Failing to use the proper player version will result in your Flash project "breaking" when run.

Adding the Slideshow Code in Flash

You are now ready to add the slideshow code. It is important to create a separate MovieClip which is used exclusively to hold the slideshow; name it something like `slideshowHolder`. Treat this movieclip as if it were the slideshow itself when positioning. Don't add any other content to this clip; it should be used exclusively for the slideshow. Open this MovieClip for editing and switch to the Actions panel (Window > Actions). Add the following code to the frame:

```
// Make sure we're stopped
stop();

// Only create the slideshow if it hasn't already been created
if (ssCreated !== true)
{
        // Update the created flag
        var ssCreated:Boolean = true;

        // Create the slideshow
        var ssParams:Object = INITIALIZATION_PARAMETER_OBJECT_HERE;
        var ssLoader:XFSLoader = new XFSLoader(this, ssParams);
        ssLoader.addEventListener(XFSLoader.LOADING_ERROR, handleSSError);
        function handleSSError(event:Event):void
        {
                trace("Error: Unable to load slideshow.");
        }
}
```

You must replace the `INITIALIZATION_PARAMETER_OBJECT_HERE` value with the actual initialization code you earlier located in the slideshow HTML code. When updated, the line should look something like this:

```
var ssParams:Object = {width:'300', height:'200', level:'a', xml:'v4flashslideshow/slideshow_data1.xml',
backgroundColor:'#000000', backgroundAlpha:'1', preloaderTextColor:'#999999'};
```

If you make changes to your slideshow in Dreamweaver which affect the initialization code, you'll need to update it in your Flash project as well.

In a nutshell, the code first stops the movieclip and checks that the slideshow hasn't already been created. This prevents you from accidentally creating multiple instances of the same slideshow. Next, the slideshow parameter object is defined. This is passed to a new slideshow loader object (an instance of the XFSLoader class). Also passed is the movieclip holder to be used, which is simply `this`. Finally, an event handler is added to provide a trace message if the loading fails.

It's really as simple as that. If you want to do some more advanced stuff (such as interact with the slideshow), see below. As mentioned above, just make sure to upload all of the slideshow-related files in addition to your SWF file when deploying.

Preview Issues

If your FLA project file is in a different folder than the deployment location, you won't be able to preview the slideshow within the Flash authoring tool. You'll receive an error message about the slideshow not being able to load.

Additionally, when previewing locally you may receive a Flash security error message indicating that the file cannot be loaded. To fix this, you must add all of the SWF files being used (both your project SWF's as well as the v4flashslideshow/ supporting files folder) as trusted local content in the Flash Player security settings dialog. To learn more about doing this, see this topic.

Advanced Information and Techniques

Please note: This section is only for users familiar with AS3.

The `XFSLoader` class is defined in the XFSLoader.swc file which was added to your project earlier. It extends `EventDispatcher` and is never added to the display list; instead, the holder display object is passed to its constuctor.   The constructor for this class takes two parameters:

- `parentHolder` - This is the display object container (Sprite, MovieClip, etc) which will be taken over and used for the slideshow.   This needs to be a dedicated holder; no other contents should be placed in it.   This holder **must** be on the stage when the XFSLoader instance is created.
- `params` - This is an object which represents the slideshow initialization parameters.   This object can be taken directly from the slideshow HTML as shown above.

There are several special things to note regarding the parameter object.   First, the path to the appropriate SWF file is automatically determined using the path to the XML configuration file.   If you have moved files around such that this path is not correct, you can manually specify the relative path to the appropriate SWF file (relative to the HTML page which will hold the Flash project) by adding an explicitly-defined `swf` parameter in the initialization object. Second, the slideshow is automatically masked to prevent unwanted overflow of the contents.   If you wish to disable this mask, add a `useMask:'false'` parameter to the initialization object.

Once you have created the instance of the XFSLoader class, you can add event listeners for the following events which it dispatches:

- `XFSLoader.LOADING_ERROR` - This event type constant defines the event dispatched when a loading error occurs which prevents the slideshow from being displayed. When errors occur, more detailed information will usually be traced out to the Output panel as well.
- `XFSLoader.SS_READY` - This event type constant defines the event dispatched once the slideshow is loaded and may be accessed via the public *`xfsLoaderInstanceName`*`.`**`ss`** property (see below for more information).

Accessing the Loaded Slideshow

Once the `SS_READY` event has been dispatched, you can access the slideshow via the `ss` property on the XFSLoader instance.   This provides a direct reference to the `com.dwuser.ss4.ui.FlashSlideshow` instance; this class is comprehensively documented in the API Reference. Note that this property is only available after the event has been dispatched; prior to that, it is `null`.

You can use the reference for adding event listeners or calling methods.   The following example code shows both of these.   First, a listener is added which displays the image XML each time a new image is synced. Then, a function is created which toggles autoplay when a button in the parent movieclip named `play_btn` is clicked.   This code is designed to be added to the basic example shown above, within the `if (ssCreated !== true) { ... }` statement:

```
// Also add the SS_READY listener
ssLoader.addEventListener(XFSLoader.SS_READY, handleSSReady);
```

```
// A reference which will be used to point at the slideshow
var ssRef:Object;

// This tracks whether the slideshow is playing or not
var ssPlaying:Boolean = true;

function handleSSReady(event:Event):void
{
        // Set the reference
        ssRef = ssLoader.ss;

        // We'll add a listener for the image sync event
        ssLoader.ss.addEventListener("images_syncImages", handleImageSync);
}

function handleImageSync(event:Object):void
{
        // Trace out the current image's XML
        trace(event.extraInfo.image.toXMLString());
}

// This line assumes that a play_btn button was already created.
parent.getChildByName("play_btn").addEventListener(MouseEvent.CLICK, playPauseSS);

function playPauseSS(event:Event):void
{
        // Make sure that the slideshow ref has been created.
        if (ssRef)
        {
                // Toggle play/pause by calling the FlashSlideshow.startAutoplay() or FlashSlideshow.stopAutoplay() method
                ssPlaying = !ssPlaying;
                if (ssPlaying)
                        ssRef.startAutoplay();
                else
                        ssRef.stopAutoplay();
        }
}
```

The reference method shown above is the recommended approach: First, a separate `ssRef` variable is created to hold the reference to the slideshow. Then, whenever the `ssRef` is needed (e.g. when the play/pause button is clicked), an `if` statement is used to verify that `ssRef` has a value and isn't `null`.

Integrating with Exsting Flex Projects (Pro only)

The Flex integration feature allow you to use XFS v4 slideshows as a part of existing Flex projects. This feature is only available in the Pro version of the software.

Note: If you want to integrate in a Flash project, see this topic.

Prerequisites

There are several prerequisites for Flex integration:

- As mentioned above, you must be using the Pro version of the XFS v4 software.
- It is highly recommended that you have experience and some comfortability with Flex usage and Actionscript 3.0 script editing. Without this basic knowledge, it will be very difficult to create a working implementation. If you're not sure of your skills, view the Flex help or consider instead a standard slideshow insertion in Dreamweaver.
- This guide is targeted at Flex developers who use Flex Builder (aka Flash Builder) as their development environment. However, the techniques still apply for developers using only the Flex SDK. You should be using Flex Builder (Flash Builder) **3.0.2** or higher, which includes Flash Player 10 support.
- Your project should be using the Flex SDK 3.2 or higher, which includes Flash Player 10 support. You can use lower versions, but your slideshow won't be able to use the Professional SWF version (as detailed below), because it relies on Flash Player 10 features.

Creating the Slideshow / File Structure

You must use Dreamweaver to create the slideshow which will be integrated into the Flex project. First, create a separate HTML page to hold the slideshow. This page won't be viewed online; it will only be used to hold the slideshow, allowing you to make edits. **Important:** To ensure that all paths are correct and functional, it is **critical** that you save this file in the same directory as the HTML page which displays your Flex-created SWF. For example, if your main Flex file is viewed at http://example.org/index.htm, then the slideshow holder page should be something like http://example.org/slideshow_holder.htm. In Flex, you often won't be running/debugging directly from the deployment directory; instead, you'll be using a `bin-debug/` directory. In these cases, where your preview page is located in some other folder than the display folder (e.g. http://example.org/bin-debug/myProject.html in the above example), then the slideshow will not load while debugging/previewing prior to deployment. Rest assured, though, that it will appear properly when viewed in the context of the deployment file structure. You can verify this by moving the relevant `bin-debug/` files (including the main Flex HTML page) to the folder where your slideshow holder page is located and viewing the Flex HTML page from there.

When you deploy your Flex project, you will have to upload all of the slideshow-related files as detailed in the slideshow insertion portion of the manual. Failing to upload all required files will result in the slideshow not appearing.

Once the slideshow has been inserted, select it and switch to Code View in Dreamweaver. Locate the initialization object where the slideshow HTML parameters are passed. This is the line which appears similar to the following:

```
XMLFlashSlideshow_v4({width:'300', height:'200', level:'a', xml:'v4flashslideshow/slideshow_data1.xml',
backgroundColor:'#000000', backgroundAlpha:'1', preloaderTextColor:'#999999'});
```

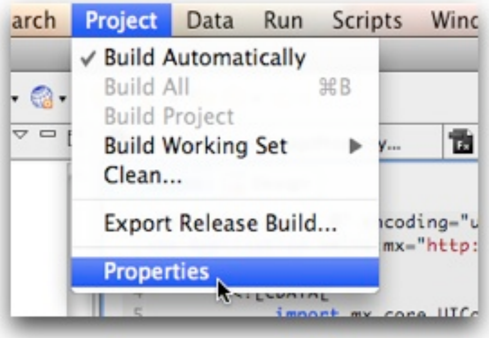
Note the bolded code, which starts with { and ends with } (the curly braces).   This is the actual initialization object which passes important information such as the dimensions, the XML configuration file path, background color, etc.   This is the slideshow parameter object which you will use later. When you make changes to the slideshow settings in the wizard, this parameter object may change, requiring you to update it in your Flex project (see below).

Configuring the Flex Project

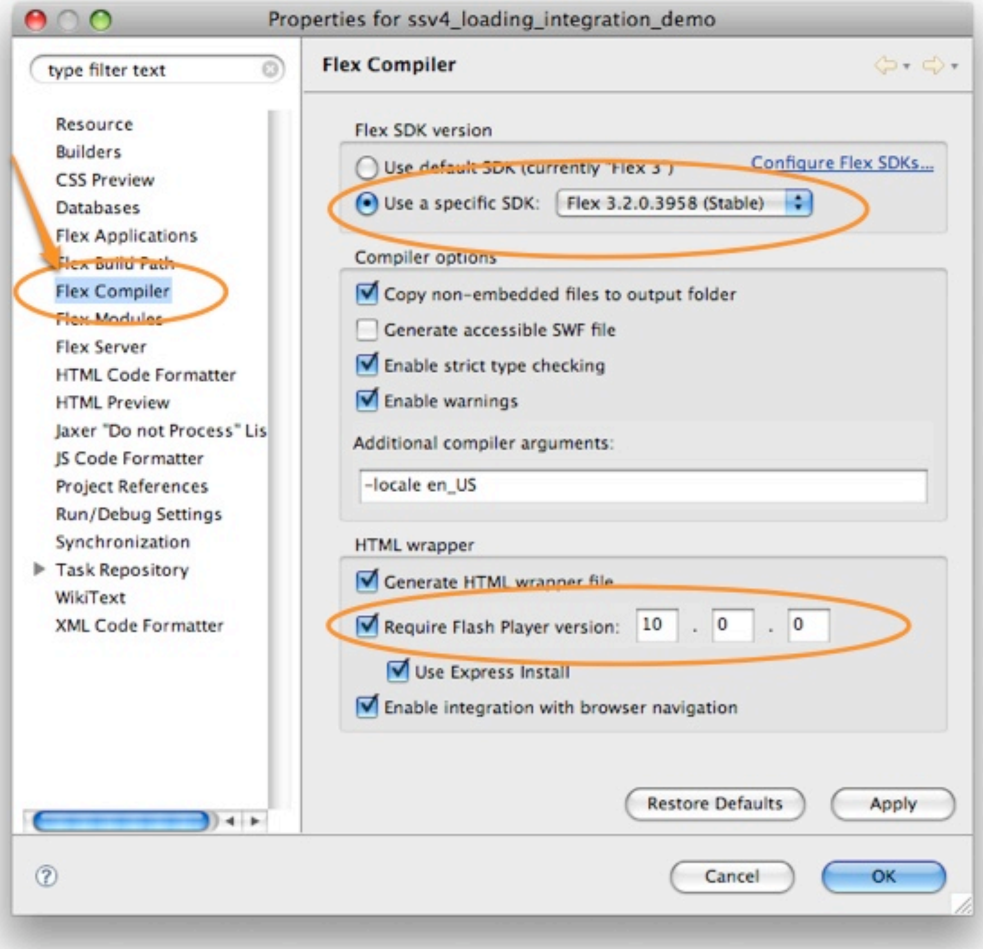
First, you need to download the `XFSLoader` SWC file which contains the special slideshow loader class.   The `XFSLoader.swc` file is available here. Download it and place it in your Flex Builder project's `libs/` directory; this will ensure its automatic inclusion.

Next, you need to configure the appropriate Flash Player version.   This setting is dependent on the SWF version you specified in the 'Slideshow Type - Pro or Basic' pod in the Global Configuration and Playback Settings section of the main wizard when creating the slideshow.   If you chose the Basic SWF, you can use the Flash Player 9 or higher.   If you chose the Professional SWF, you **must** use the Flash Player 10 or higher.   Failing to use the proper player version will result in your Flash project "breaking" when run.

To set the required Flash Player version, open the Project Properties dialog by selecting Project > Properties in Flex Builder:



In the Properties dialog, select the Flex Compiler section on the left side:



Properties for ssv4_loading_integration_demo

Flex Compiler

Flex SDK version

○ Use default SDK (currently "Flex 3")

● Use a specific SDK: Flex 3.2.0.3958 (Stable)

Compiler options

☑ Copy non-embedded files to output folder

□ Generate accessible SWF file

☑ Enable strict type checking

☑ Enable warnings

Additional compiler arguments:

-locale en_US

HTML wrapper

☑ Generate HTML wrapper file

☑ Require Flash Player version: 10 . 0 . 0

☑ Use Express Install

☑ Enable integration with browser navigation

Note: As mentioned above, this guide assumes (and highly recommends) that you are using the Flex SDK 3.2 or higher. This is required if you wish to target the Flash Player 10.

To set the Flash Player version, enter a value in the 'Require Flash Player version' box. For the Professional SWF, you should use 10.0.22 (short for 10.0.22.87) or higher; for the Basic SWF, you should use 9.0.124. While you could use 10.0.0 and 9.0.0, both of the listed point-release versions include important security updates.

Note: If you are not using the Generate HTML Wrapper file option (resulting in a grayed-out 'Require Flash Player version' box) and need to force the Flash Player 10 to be used, add `-target-player=10.0.22` in the 'Additional Compiler Arguments' field.

Adding the Slideshow Code in an MXML File

You are now ready to add the slideshow code. Because all direct Flex display children must be `UIComponent` instances, the slideshow must be wrapped in a specific UIComponent holder, which is used exclusively to hold the slideshow. Open the Flex component Flex application where you wish to include the slideshow and add the following line to your MXML code:

```
<mx:UIComponent id="slideshowHolder" x="100" y="100" addedToStage="createSlideshow()" />
```

This example uses an `id` value of `slideshowHolder` and it is positioned at 100,100. The position can be set to whatever you wish, but you should not explicitly specify dimensions; the dimensions are passed on the slideshow initialization object. The yet-uncreated `createSlideshow()` function is set to be called when the holder is added to the stage. It is important to note that the holder **must** be on the stage prior to any slideshow loading being initialized; doing otherwise will result in the slideshow failing with an error message. This is why the `addedToStage` event is used.

Next, you need to define the `createSlideshow` function and its supporting code by adding the following `<mx:Script>` block at the top of your MXML file:

```
<mx:Script>
        <![CDATA[

                private var ssLoader:XFSLoader;
                private var ssRef:Object;

                private function createSlideshow():void
                {
                        // Create the slideshow using the holder and add a loading error listener
                        var ssParams:Object = INITIALIZATION_PARAMETER_OBJECT_HERE;
                        ssLoader = new XFSLoader(slideshowHolder, ssParams);
                        ssLoader.addEventListener(XFSLoader.LOADING_ERROR, handleSSLoadError);
                }

                private function handleSSLoadError(event:Event):void
                {
                        trace("Error loading slideshow!");
                }

        ]]>
</mx:Script>
```

You must replace the `INITIALIZATION_PARAMETER_OBJECT_HERE` value with the actual initialization code you earlier located in the slideshow HTML code. When updated, the line should look something like this:

```
var ssParams:Object = {width:'300', height:'200', level:'a', xml:'v4flashslideshow/slideshow_data1.xml',
backgroundColor:'#000000', backgroundAlpha:'1', preloaderTextColor:'#999999'};
```

If you make changes to your slideshow in Dreamweaver which affect the initialization code, you'll need to update it in your Flex project as well.

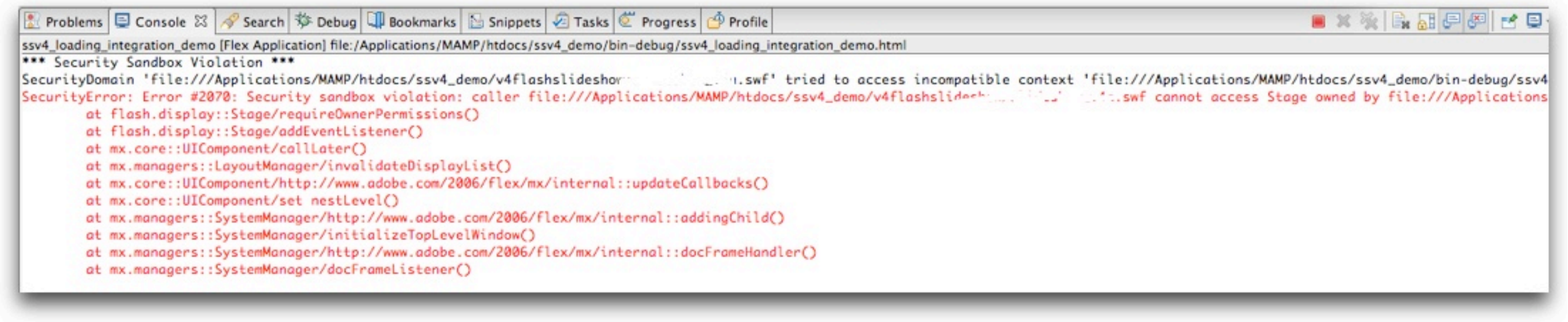
In a nutshell, the code first creates variables to hold both the slideshow loader instance and the slideshow reference. Next, the createSlideshow function is defined. It defines the slideshow parameter object, which is passed to a new slideshow loader object (an instance of the XFSLoader class). Also passed is the holder to be used, which is the previously created `slideshowHolder` UIComponent instance. Finally, an event handler is added to provide a trace message if the loading fails.

It's really as simple as that. If you want to do some more advanced stuff (such as interact with the slideshow), see below. As mentioned above, just make sure to upload all of the slideshow-related files in addition to your Flex SWF file when deploying.

Preview Issues

If your Flex preview/debug file is in a different folder than the deployment location, you won't be able to preview the slideshow when testing within Flex builder -- you'll receive an error message about the slideshow not being able to load. Rest assured, though, that the slideshow will appear properly when viewed in the context of the deployment file structure. For more information and how to verify paths while still developing, see the File Structure section above.

Additionally, when previewing locally you may receive a Flash Player security error message indicating that the file cannot be loaded, similar to this:



```
ssv4_loading_integration_demo [Flex Application] file:/Applications/MAMP/htdocs/ssv4_demo/bin-debug/ssv4_loading_integration_demo.html
*** Security Sandbox Violation ***
SecurityDomain 'file:///Applications/MAMP/htdocs/ssv4_demo/v4flashslideshow[illegible].swf' tried to access incompatible context 'file:///Applications/MAMP/htdocs/ssv4_demo/bin-debug/ssv4
SecurityError: Error #2070: Security sandbox violation: caller file:///Applications/MAMP/htdocs/ssv4_demo/v4flashslid[illegible].swf cannot access Stage owned by file:///Applications
	at flash.display::Stage/requireOwnerPermissions()
	at flash.display::Stage/addEventListener()
	at mx.core::UIComponent/callLater()
	at mx.managers::LayoutManager/invalidateDisplayList()
	at mx.core::UIComponent/http://www.adobe.com/2006/flex/mx/internal::updateCallbacks()
	at mx.core::UIComponent/set nestLevel()
	at mx.managers::SystemManager/http://www.adobe.com/2006/flex/mx/internal::addingChild()
	at mx.managers::SystemManager/initializeTopLevelWindow()
	at mx.managers::SystemManager/http://www.adobe.com/2006/flex/mx/internal::docFrameHandler()
	at mx.managers::SystemManager/docFrameListener()
```

To fix this, you must add all of the SWF files being used (both your project SWF's as well as the v4flashslideshow/ supporting files folder) as trusted local content in the Flash Player security settings dialog. To learn more about doing this, see this topic.

Advanced Information and Techniques

Please note: This section is only for users familiar with AS3.

The `XFSLoader` class is defined in the XFSLoader.swc file which was added to your project earlier. It extends `EventDispatcher` and is never added to the display list; instead, the holder display object is passed to its constuctor.   The constructor for this class takes two parameters:

- `parentHolder` - This is the display object container (UIComponent, Sprite, MovieClip, etc) which will be taken over and used for the slideshow.   This needs to be a dedicated holder; no other contents should be placed in it.
- `params` - This is an object which represents the slideshow initialization parameters.   This object can be taken directly from the slideshow HTML as shown above.

There are several special things to note regarding the parameter object.   First, the path to the appropriate SWF file is automatically determined using the path to the XML configuration file.   If you have moved files around such that this path is not correct, you can manually specify the relative path to the appropriate SWF file (relative to the HTML page which will hold the Flex SWF) by adding an explicitly-defined `swf` parameter in the initialization object. Second, the slideshow is automatically masked to prevent unwanted overflow of the contents.   If you wish to disable this mask, add a `useMask:'false'` parameter to the initialization object.

Once you have created the instance of the XFSLoader class, you can add event listeners for the following events which it dispatches:

- `XFSLoader.LOADING_ERROR` - This event type constant defines the event dispatched when a loading error occurs which prevents the slideshow from being displayed. When errors occur, more detailed information will usually be traced out to the Output panel as well.
- `XFSLoader.SS_READY` - This event type constant defines the event dispatched once the slideshow is loaded and may be accessed via the public *xfsLoaderInstanceName*.**ss** property (see below for more information).

Accessing the Loaded Slideshow

Once the `SS_READY` event has been dispatched, you can access the slideshow via the `ss` property on the XFSLoader instance.   This provides a direct reference to the `com.dwuser.ss4.ui.FlashSlideshow` instance; this class is comprehensively documented in the API Reference. Note that this property is only available after the event has been dispatched; prior to that, it is `null`.

You can use the reference for adding event listeners or calling methods.   The following example code shows both of these.   First, a listener is added which displays the image XML each time a new image is synced. Then, a function is created which toggles autoplay when a separately-created button is clicked. The following complete detailed example builds on the basic example shown above; again, the `INITIALIZATION_PARAMETER_OBJECT_HERE` value must be replaced with the actual initialization code you earlier located in the slideshow HTML code:

```
<mx:Script>
        <![CDATA[

                private var ssLoader:XFSLoader;
                private var ssRef:Object;

                private function createSlideshow():void
                {
                        // Create the slideshow using the holder and add a loading error listener
                        var ssParams:Object = INITIALIZATION_PARAMETER_OBJECT_HERE;
                        ssLoader = new XFSLoader(slideshowHolder, ssParams);
                        ssLoader.addEventListener(XFSLoader.LOADING_ERROR, handleSSLoadError);
```

```
            ssLoader.addEventListener(XFSLoader.SS_READY, handleSSReady);
        }

        private function handleSSLoadError(event:Event):void
        {
            trace("Error loading slideshow!");
        }

        private function handleSSReady(event:Event):void
        {
            // Create the reference to the slideshow
            ssRef = ssLoader.ss;

            // Use the ref to add a listener for the image sync event
            ssRef.addEventListener("images_syncImages", handleImageSync);
        }

        private function handleImageSync(event:Object):void
        {
            // This example traces out the XML for each image as it is displayed
            trace(event.extraInfo.image.toXMLString());
        }

        // Create a flag which tracks whether the slideshow is playing or not
        private var ssPlaying:Boolean = true;

        private function toggleAutoplay():void
        {
            // Make sure that the slideshow ref has been created.
            if (ssRef)
            {
                // Toggle play/pause by calling the FlashSlideshow.startAutoplay() or FlashSlideshow.stopAutoplay() method
                ssPlaying = !ssPlaying;
                if (ssPlaying)
                    ssRef.startAutoplay();
                else
                    ssRef.stopAutoplay();
            }
        }

    ]]>
</mx:Script>

<mx:UIComponent id="slideshowHolder" x="100" y="100" addedToStage="createSlideshow()" />

<mx:Button x="100" y="340" label="Play / Pause" click="toggleAutoplay()" />
```

The reference method shown above is the recommended approach: First, a separate `ssRef` variable is created to hold the reference to the slideshow. Then, whenever the `ssRef` is needed (e.g. when the play/pause button is clicked), an `if` statement is used to verify that `ssRef` has a value and isn't `null`.

Migrating Slideshows from v3 to v4

Many aspects of the slideshow configuration were changed greatly between version 3 and version 4. This makes it difficult to transition an entire slideshow instantly from version 3 to version 4. However, one aspect of the slideshow configuration structure which was extended rather than revamped is the <galleries> section of the XML file, where all image galleries are defined. The result is that the image data from a version 3 slideshow can be easily transitioned to a version 4 slideshow.

As a side note, it should be noted that version 3 and version 4 of the extension are fully interoperable. It is not necessary to uninstall version 3 prior to installing version 4. In fact, you can use version 3 and 4 slideshows together on the same site (using them on the same page is not recommended).

To perform a slideshow data migration, first open the version 3 slideshow's XML configuration file. This can be done in Dreamweaver by selecting the slideshow in Design View and clicking Edit Manually in the Properties panel (Window > Properties). Next, place your cursor next to the v3 slideshow and run the v4 Insert Wizard. **Don't spend time configuring the slideshow yet** - just add a single empty gallery, then save and exit. Select the newly-inserted v4 slideshow and click the Edit Manually... button in the Properties panel to open its XML configuration file as well.

You should now have the v3 and v4 configuration files open side by side. Near the bottom of the v3 file, locate the `<galleries ... > ... </galleries>` section, which will appear something like this, with one or more `<gallery>` child nodes:

```
  <galleries ... >

        <gallery thumb="images/ss_images/water_sky/s_213400_8302.jpg" title="H2o : water" description="Seascapes, shorelines and more, all featurir
              <img src="213400_8302.jpg" />
              <img src="543909_24516987.jpg" />
              <img src="465818_34780293.jpg" />
              <img src="465866_40544776.jpg" />
              <img src="472069_32282489.jpg" />
              <img src="479980_27510256.jpg" />
        </gallery>

        <gallery thumb="images/ss_images/autumn/s_298562_6417.jpg" title="Autumn" description="Fascinating colors as nature makes its transition f
              <img src="468871_85843609.jpg" />
              <img src="469009_10111833.jpg" />
              <img src="475160_97624508.jpg" />
              <img src="483503_26439024.jpg" />
```

```
            <img src="485916_64203504.jpg" />
            <img src="488361_60127403.jpg" />
            <img src="493993_21435017.jpg" />
            <img src="496416_69929600.jpg" />
        </gallery>

    </galleries>
```

Copy this code (including `<galleries ... >` and `</galleries>` !) and switch to the v4 configuration file.   Locate the corresponding `<galleries ... > ... </galleries>` section at the bottom of the v4 file and **replace it** with the contents you copied from the v3 file. Save your change and close both configuration files.

Now that the gallery and image data has been migrated to the version 4 slideshow, you can switch back to the HTML page and launch the edit wizard for the version 4 slideshow.   (If you receive XML parsing errors, close the edit wizard, re-open the v4 configuration file, and make sure you didn't make any errors when replacing the code.) You can now configure and customize the slideshow as desired; all of the migrated galleries should appear in the Image Galleries tab.   You can customize the migrated images with version-4-specific options as well.

Important: Once you're ready to remove the old version 3 slideshow, be sure to also remove any and all `<script src="v3flashslideshow/slideshow.js" ... ></script>` file references from your file to prevent conflicts.   Failing to do this may could cause version 4 slideshows to not appear properly.

Important Note: The format for the *autoplayDelay* attribute changed between version 3 and 4. If your copied v3 slideshow code included a `<gallery ... autoplayDelay="..." ... >` value, remove it and then re-create your autoplay delay in the Global Config / Playback Settings section of the main wizard.

Advanced Topics / Advanced Examples > **Preloading the SWF File**

Preloading the SWF File

In order to provide the fastest possible load time to your users, you can preload the SWF file on pages which don't have slideshows. As users browse your site, the SWF will quietly load in the background. Then, when the users visit page(s) with slideshows, the slideshow will appear instantly because the SWF file will be stored in the user's temporary internet cache. After the slideshow SWF file has been loaded or preloaded once, it will appear instantly on all subsequent requests. The method shown below will not affect the performance of your other pages in any way; all loading takes place after the rest of the document has loaded, while the user is browsing the page.

This method should generally only be used on sites where slideshows are a central feature of the site and most users will view one. If users don't ever view a slideshow, preloading the slideshow SWF will use unnecessary bandwidth on your server.

Creating the Special Preloader Page

To setup preloading, first create a new special HTML page. Save it as something like `slideshow_preloader.html` , putting it in an easily accessible directory. It is best if you save it in the same folder as the pages where you want to add preloading capabilities. In this page, insert a new slideshow. Only add a single empty image gallery, and set the slideshow dimensions to 10 x 10 (or some other very small size). Set the 'Slideshow Type - Pro or Basic' setting in the Global Configuration and Playback Settings section to be the same as your other slideshows. If you use both types, insert a second empty slideshow which uses the other Slideshow Type setting.

Enabling Preloading on an Existing Page

Important: Only enable preloading on pages without slideshows. For example, if your site has a 'home' page and a 'galleries' page, you would place the preloading page only on the 'home' page.

Open the page to which you want to add preloading and switch to code view. At the end of the document, just before the closing `</body>` tag, add the following code:

```
<!-- BEGIN XFS v4 Preload Script -->
    <div style="display: none">
        <iframe id="xfs4PreloadFrame" width="1" height="1" frameborder="0" scrolling="no"></iframe>
        <script language="javascript" type="text/javascript">
        function addLoadEvent(func) { var oldonload = window.onload; if (typeof window.onload != 'function') { window.onload = func; } else {
        addLoadEvent(function() {
            var preloadPage = "RELATIVE_PATH_TO_PRELOAD_PAGE_HERE";
            var iframe = document.getElementById('xfs4PreloadFrame');
            if (iframe)
                iframe.src = preloadPage;
        });
        </script>
    </div>
<!-- END XFS v4 Preload Script -->
```

You must replace the `RELATIVE_PATH_TO_PRELOAD_PAGE_HERE` code with the actual relative path from the current page to the earlier-created preloader page. The line should look something like this after being updated:

```
var preloadPage = "slideshow_preloader.html";
```

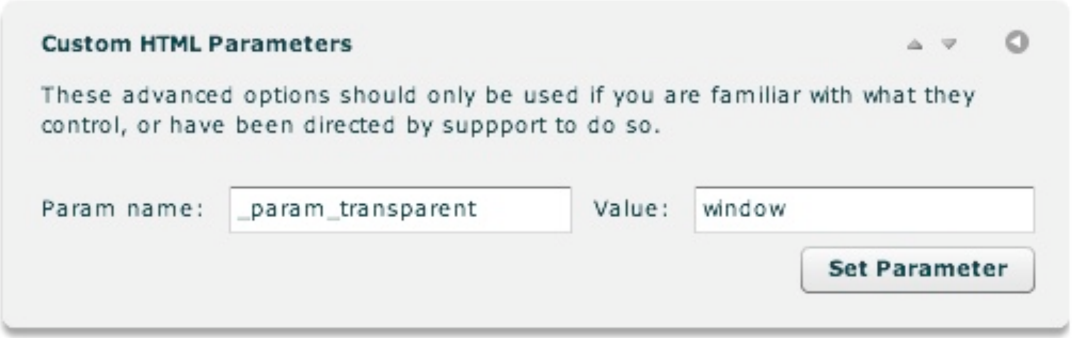
Now, once the page finishes loading and the user is browsing, the slideshow SWF file(s) will load silently in the background. You can add the preloading code to only one page, or many pages -- just make sure to specify the correct path to the preloader page each time.

Tips for Using 3D Image Displays

When using the 3D image display elements in the Pro version of the software, the following tips can help improve your users' experience.

Disabling Transparency (Boosts Performance)

If you're not actively taking advantage of the slideshow's support for a transparent background, you should disable transparency when using a 3D image display element. This will allow the slideshow to render more quickly, boosting performance. To disable transparency, the `wmode` SWF embed parameter must be set to `window` . This can be done in the Custom HTML Parameters pod in the Global Configuration and Playback Settings section. The Param Name must be set to `_param_wmode` and the value set to `window` :



Custom HTML Parameters

These advanced options should only be used if you are familiar with what they control, or have been directed by suppport to do so.

Param name: _param_transparent Value: window

Using Slide-Based Events

Note: This is an advanced topic only for users who are familar with XML and Actionscript editing. Image events are explained in more detail in the `com.dwuser.ss4.ui.IndividualSlide` class in the API Reference.

Slide-based events are a set of special events which are dispatched by each slide. Handlers for these events are added directly to the <img> node which represents each slide. These events include:

- **The Sync Event** - Handled via the `event_sync` XML attribute.
  Dispatched when the slide finishes transitioning in (opening) and is synced. Referencing `event.`***`someValue`*** in the handler allows for access to the original `SlideshowEvent` event properties, such as `extraInfo` (see `SlideshowEvent.IMAGES_SYNCIMAGES` in the API Reference for more info). Referencing `event.currentTarget.`***`someValue`*** is the same as referencing `slideshow.`***`someValue`*** , since the `SlideshowEvent` is dispatched from the main slideshow.

- **The Click Event** - Handled via the `event_click` XML attribute .
  Dispatched when the slide is clicked. Referencing `event.`***`someValue`*** in the handler allows for access to the original `MouseEvent` event properties. Referencing `event.currentTarget.`***`someValue`*** is the same as referencing `currentSlideReference.`***`someValue`*** , since the `MouseEvent` is dispatched from the slide in question. In other words, the event's `currentTarget` property provides a reference to the dispatching `IndividualSlide` instance.

- **The Roll-over Event** - Handled via the `event_rollOver` XML attribute.
  Dispatched when the slide is moused over. Referencing `event.`***`someValue`*** in the handler allows for access to the original `MouseEvent` event properties. Referencing `event.currentTarget.`***`someValue`*** is the same as referencing `currentSlideReference.`***`someValue`*** , since the `MouseEvent` is dispatched from the slide in question. In other words, the event's `currentTarget` property provides a reference to the dispatching `IndividualSlide` instance.

- **The Roll-out event** - Handled via the `event_rollOut` XML attribute.
  Dispatched when the user mouses off the slide. Referencing `event.`***`someValue`*** in the handler allows for access to the original `MouseEvent` event properties. Referencing `event.currentTarget.`***`someValue`*** is the same as referencing `currentSlideReference.`***`someValue`***, since the `MouseEvent` is dispatched from the slide in question. In other words, the event's `currentTarget` property provides a reference to the dispatching `IndividualSlide` instance.

Handlers added for any of these events will be called when the corresponding event occurs. This allows you to dispatch a global event or perform some other task when a local slide event occurs. As mentioned above, the handlers are added directly to the <img> node which represents each slide. It is best to illustrate this with some example code:

Example Handler 1 - The following listener starts autoplay when the image is opened / synced:

```
<img ... event_sync="event.currentTarget.startAutoplay();" />
```

Example Handler 2 - The following listener calls the javascript alert function (which displays an alert box) when the slide is rolled over:

```
<img ... event_rollOver="navigateToURL(new URLRequest('javascript:alert(\'You just rolled over the
slide!\')'));" />
```

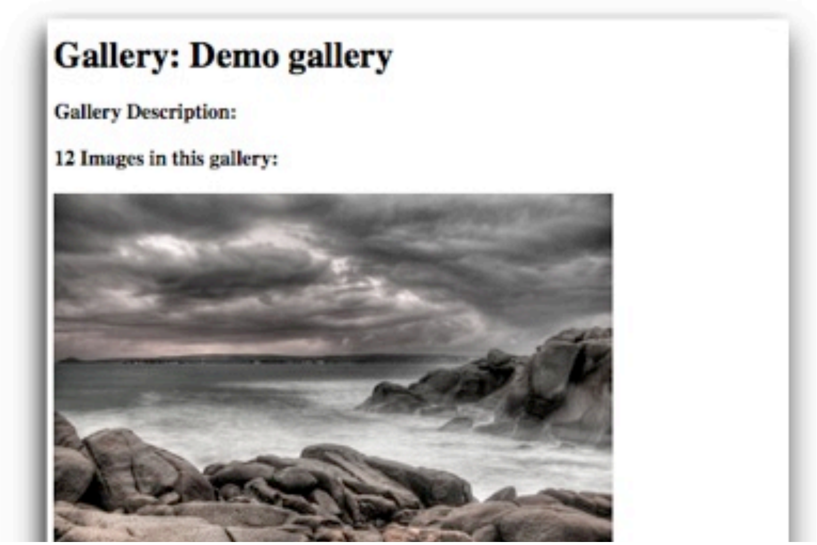
The most common use (as shown above) is to start autoplay when a specific image is opened. As a more complex example, you might fade out another portion of the slideshow UI when the user hovers over an image, then restore it when the hover out. These events are especially helpful when integrating a slideshow in a custom Flash or Flex project.

Accessibility / Section 508 Compliance

Due to the dynamic Flash-based display of the slideshows, they are not generally usable with screenreaders or for users who need other accessibility provisions. The solution to this is to provide an alternate page which contains the HTML equivalent of the slideshow's contents. Then, provide a small unobtrusive link below the slideshow which says something like "Accessible version" or "Text version". This method is used by a number of high-profile websites, such as the CIA.gov site.

The task of converting a slideshow's contents into an alternative HTML equivalent can be daunting. The following tool simplifies this process by generating the content automatically from your slideshow's XML configuration file. To use the tool, first open your slideshow's XML configuration file by selecting it in Design View and selecting the Edit Manually... button in the Properties panel (Window > Properties). Paste this code into the box below, then click the convert button. Be patient -- the processing may take a few seconds for larger slideshows. You can use the 'Display Images in Alternate Content' option to specify whether the alternate content should be text-only or should also include the photos. The lower box will then display equivalent HTML contents. Create a new HTML page **in the same folder as the original slideshow's HTML page**, naming it something like `slideshow_accessible.html` . Between the `<body> ... </body>` tags, paste the generated HTML content. It is important to save the file in the same folder as the original slideshow's HTML page; failing to do this will result in the paths in the alternate file being broken.

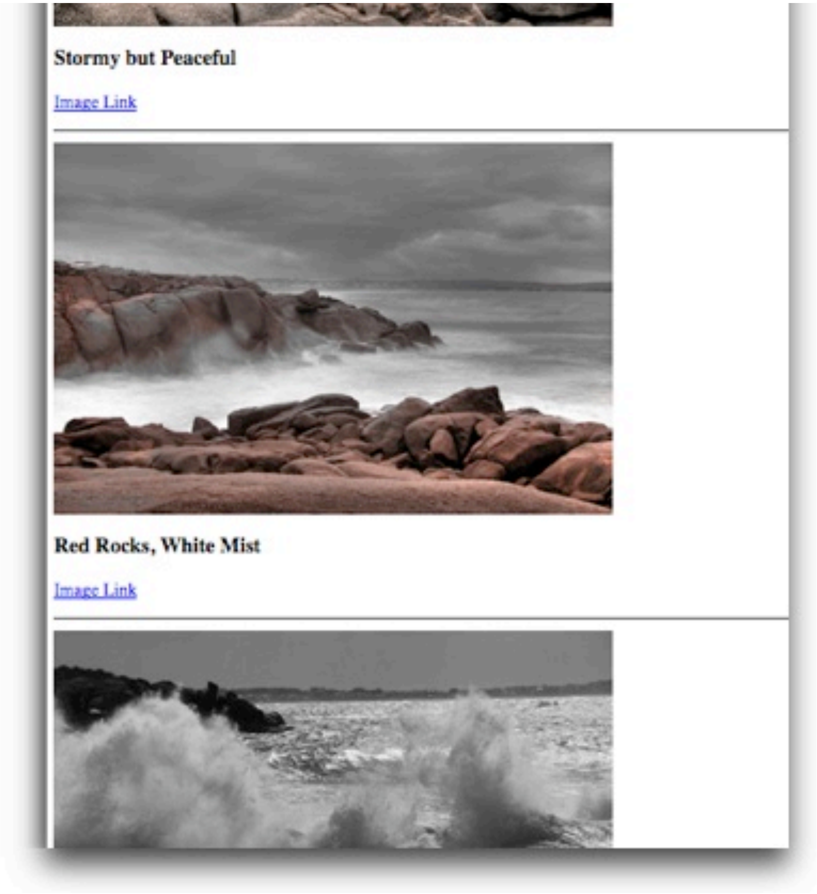
The generated HTML produces a page which appears like this, listing all galleries and their images:



Gallery: Demo gallery

Gallery Description:

12 Images in this gallery:



Stormy but Peaceful

Image Link

Red Rocks, White Mist

Image Link

Accessibility Conversion Tool

Paste the XML Configuration File Contents:

Display Images in Alternate Content? ☑

Convert *Be patient - it may take a few seconds for large galleries.*

Output - Paste into Alternate Content File:

Moving Existing Slideshows

Once you have inserted a slideshow into a page, it is generally recommended that you don't move that page to a different directory. In cases where you must move the file to a different directory, special steps must be taken to ensure that the slideshow continues to display properly. If you are not familiar with relative paths, you should view this helpful article before continuing.

Necessary Steps

It might seem obvious, but first make sure that the slideshow actually is broken. Depending on where your files are relative to each other, modifications may not be needed. If you receive a message reading *To use the XML Flash Slideshow v4, you must upload the v4flashslideshow/slideshow.js file, and have a properly defined <script> reference to it in your HTML*, then you must repair the slideshow file paths.

First, the reference to the `v4flashslideshow/slideshow.js` file must be updated. This will be found just above the slideshow code, in a line which appears as follows:

```
<!-- Do not remove the line below!!! It is required for the DWUser XML Flash Slideshow v4. -->
<script type="text/javascript" src="v4flashslideshow/slideshow.js"></script>
```

The bolded reference must be updated to point to the new relative location of the `slideshow.js` file within the `v4flashslideshow/` supporting files folder.

Second, the reference to the XML configuration file must be updated. This will be found within the initialization parameter object, and will appear similarly to this:

```
XMLFlashSlideshow_v4({width:'300', height:'200', xml:'v4flashslideshow/slideshow_data1.xml',
backgroundColor:'#000000', ...
```

The bolded reference must be updated to point to the new relative location of the `v4flashslideshow/` supporting files folder. If you made the changes correctly, you'll be able to select the slideshow in Design View and click the *Edit Manually* button in the property inspector to open the XML file for manual editing.

Third, you must update the image paths in the XML configuration file. The paths must be updated to point to the images **relative to the HTML page**. If your images are all in a single folder, this is a relatively simple operation. In the XML file, locate each `<gallery>` node in the `<galleries>` section, each of which will look similar to this:

```
<gallery title="Demo gallery" description="Description Text" fullPathPrefix="images/rocks/">
```

The bolded `fullPathPrefix` value represents the path prepended to every image's file name. It is the relative path from the HTML to the images folder. This must be updated to reflect the new location of the HTML page. **Important:** If there is a `thumbPathPrefix` value, update that as well. It should point to the thumbnails folder.

Tip: To learn more about the XML configuration file format, see the Developer Reference.

Fourth, if applicable, use the wizard to update any paths to video, audio, or external skin files. (You can make these changes manually, but it is generally easier to use the wizard.) Essentially, the paths to all external files must be updated so that they reflect the new relative path from the HTML page to the asset.

Debugging

If, after making the above changes, you still have trouble, check the following:

- If you still receive a message reading *To use the XML Flash Slideshow v4, you must upload the v4flashslideshow/slideshow.js file, and have a properly defined <script> reference to it in your HTML*, then check that the <script> reference to the `v4flashslideshow/slideshow.js` file is correct.
- If the slideshow appears but displays an XML loading error, check that you correctly updated the path to the XML file.
- If the slideshow appears properly but all images are blank and the red *Image / Asset Loading Error* bar appears at the bottom of the slideshow, check that you modified the image paths in the XML configuration file correctly.

Understanding Flash Player Cross-domain Loading Restrictions

If you are loading image files (or video or audio) or external image data from a server at a different domain, you will be affected by Flash Player Security Restrictions. In a nutshell: the Flash Player will not allow content to be loaded from a different domain than the residing SWF file sits on, unless the owner of that different domain specifically authorizes this loading.

First, it's important to explain what qualifies as a 'diffent domain'. Any difference in either a subdomain or root domain qualifies a URL as being an entirely different domain. The most common example might surprise you:

You have a slideshow on http://**www.example.org**/slideshow.html. You upload some images to your site, and add them to the slidehow with a URL 'http://**example.org**/images/image1.jpg'. When you view the slideshow, the image fails to load and you receive a security restriction error message. "Huh?" you say, "I'm loading from the same site!" The issue is that the Flash Player views **example.org** and **www.example.org** as two separate domains because their subdomains aren't identical.

More intuitive examples include a slideshow at http://www.example.org/ which loads data from http://api.flickr.com/bla . Because this is a different root domain, Flash Player security restrictions go into effect.

When you attempt to load assets (XML data, images, audio, or video files) from a different domain, the Flash Player checks to see if a crossdomain policy file has been placed on the remote server. This is a file which is named **crossdomain.xml** and sits at the root of the domain's server (e.g. http://data.example.org/crossdomain.xml or http://www.dataserver.com/crossdomain.xml). This file must either allow all domains (via the use of a wildcard) or specifically allow the domain on which your slideshow resides, or the loading will not be permitted.

To prevent loading errors (as evidenced by **'Image Failed - Security'** messages and other loading errors, such as XML loading errors), you must either load from a server which specifically permits crossdomain loading (if you don't have control over the server), or place a crossdomain.xml file on the server. Here is a simple example file, which permits loading from across all subdomains on this domain:

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM
"http://www.adobe.com/xml/dtds/cross-domain-policy.dtd">

<cross-domain-policy>
<allow-access-from domain="example.com"/>
<allow-access-from domain="*.example.com"/>
</cross-domain-policy>
```

Two live example files can be viewed at: http://search.yahooapis.com/crossdomain.xml (an open policy, with no restrictions) and http://www.amazon.com/crossdomain.xml (a more restricted policy). You can learn more about creating crossdomain.xml files (crossdomain security policy files) at Adobe's site: http://www.adobe.com/devnet/articles/crossdomain_policy_file_spec.html

Advanced users who are comfortable with server-side programming can also circumvent these restrictions through the use of a server-side proxy on the same server as hosts the slideshow. For an example, see this blog post: http://blog.nelsonsilva.eu/2008/02/27/apache-proxy-for-crossdomain-problems/ .

Avoiding "Chipmunks" Audio: MP3 Formatting Guidelines

When encoding (or re-encoding) MP3 files to use with the slideshow, there are two settings which will come up: the **sample rate** and the **bitrate**. The Flash Player has inherent limitations which must be followed to ensure that the audio plays correctly. The most common issue is with the sample rate. The preferred sample rate is 44100 Hz, but 96000, 22050 and 11025 Hz are all supported. Failing to use a supported sample rate will result in the audio playing back at a distorted speed, often described as "chipmunks."

You can use any bitrate for the MP3 as long as it is constant; variable bitrates are not supported. A lower bitrate corresponds to smaller file size and faster download times, but reduces quality. A higher bitrate produces higher quality, but takes longer to load. The best bitrate depends on your situation and the type of audio you're using.

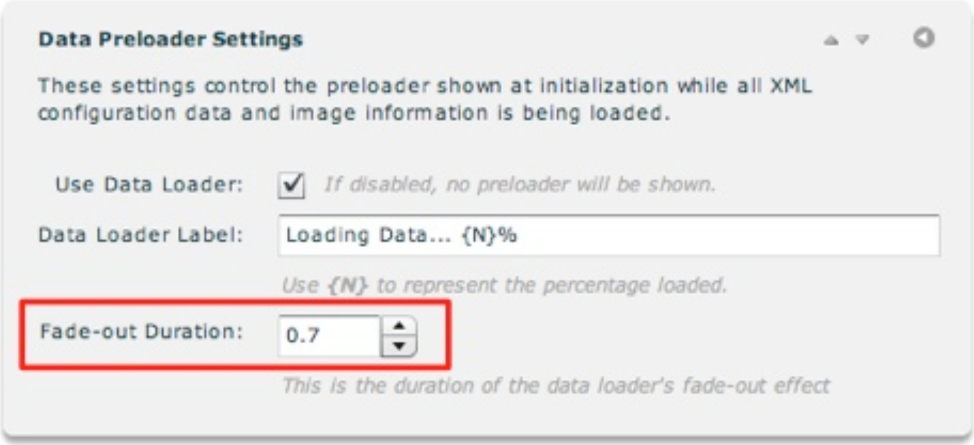
If you need to convert your audio to MP3 format or re-encode it, we recommend a freeware program called Switch. It can be downloaded for free from: http://www.nch.com.au/switch/ . Switch is compatible with both Mac and Windows.

Improving Loading Performance

The default slideshow settings result in a smooth transition from the data loader to the initial layout view. For most slideshows, this ensures an appealing visual user experience. However, for very simple slideshow configurations (e.g. image-only slideshows), the default settings can result in an unnecessary 1-2 seconds of delay prior to the first image appearing. By making some small configuration changes, this delay can be effectively removed.

There are two factors (beyond the raw transfer time required to load the XML configuration file and image files) that delay the appearance of the first photo. The first is the time used to fade out the data preloader; the second is the time used to fade in the inital layout. For image-only slideshows, the initial layout is only an empty images holder, so there is no need to fade it in.

To reduce the time used for fading out the data preloader, open the Data Preloader Settings pod in the Appearance Settings tab. Reduce the Fade-out Duration value; a value of `0` will make the data preloader (if configured to be visible) disappear instantly, or a smaller value (e.g. `0.2`) will just make it fade out much more quickly.



Data Preloader Settings

These settings control the preloader shown at initialization while all XML configuration data and image information is being loaded.

Use Data Loader: ☑ *If disabled, no preloader will be shown.*

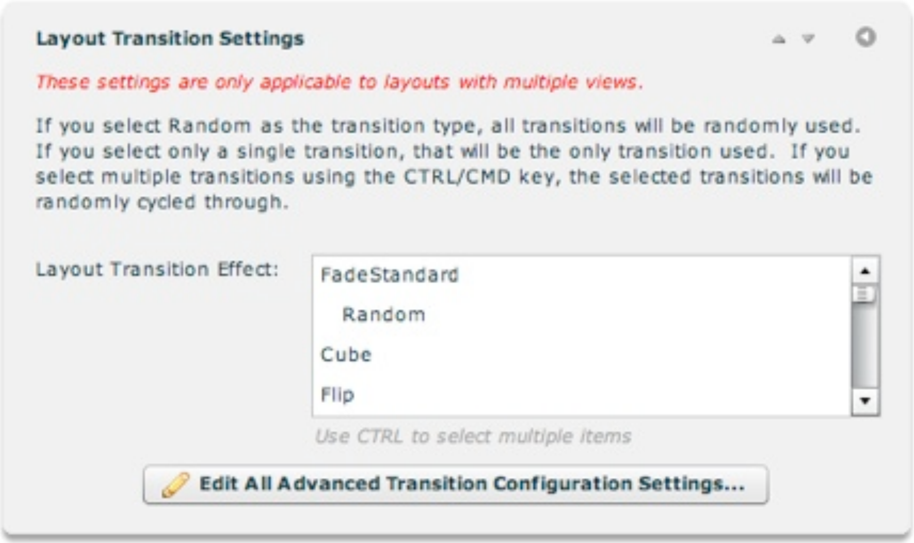
Data Loader Label: Loading Data... {N}%

Use {N} to represent the percentage loaded.

Fade-out Duration: 0.7

This is the duration of the data loader's fade-out effect

To reduce or remove the layout fade-in delay, switch to the Global Config. and Playback Settings tab and open the Layout Transition Settings pod. Click the Edit All Advanced Transition Configuration Settings button at the bottom of the pod:



Layout Transition Settings

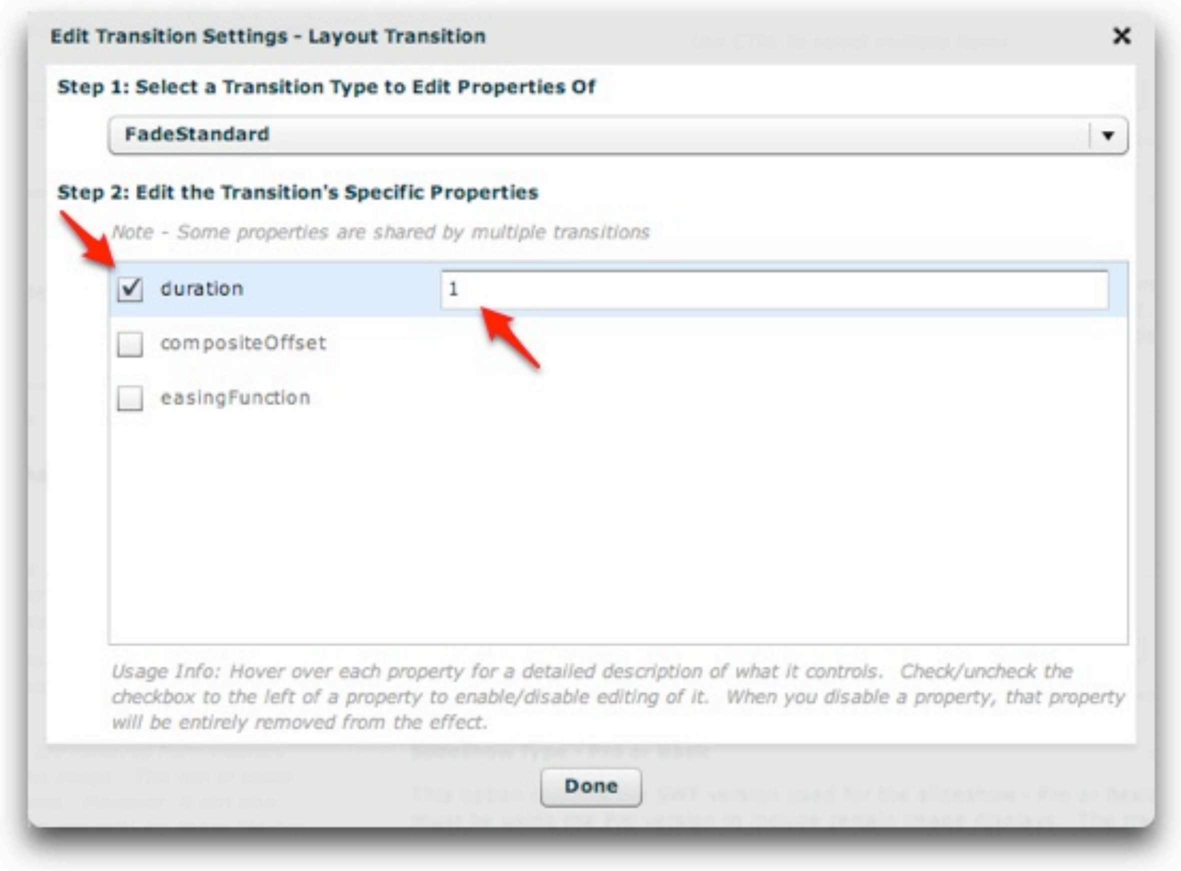
These settings are only applicable to layouts with multiple views.

If you select Random as the transition type, all transitions will be randomly used. If you select only a single transition, that will be the only transition used. If you select multiple transitions using the CTRL/CMD key, the selected transitions will be randomly cycled through.

Layout Transition Effect:

FadeStandard
Random
Cube
Flip
Use CTRL to select multiple items
Edit All Advanced Transition Configuration Settings...

The configuration dialog that launches allows for configuring the layout fade-in effect. In the dialog, check the box to enable editing of the **duration** property then enter a value of `1` in the value field. The values are denoted in milliseconds, so using `1` effectively removes the fade-in effect:



Edit Transition Settings - Layout Transition

Step 1: Select a Transition Type to Edit Properties Of

FadeStandard

Step 2: Edit the Transition's Specific Properties

Note - Some properties are shared by multiple transitions

☑ duration: 1

□ compositeOffset

□ easingFunction

Usage Info: Hover over each property for a detailed description of what it controls. Check/uncheck the checkbox to the left of a property to enable/disable editing of it. When you disable a property, that property will be entirely removed from the effect.

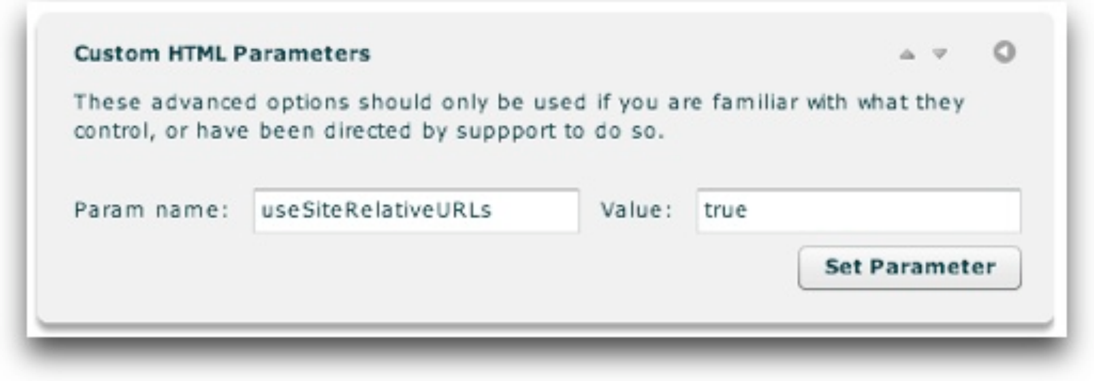
Important: Don't set the value to 0. Using a value of 0 can result in errors or a nonfunctional slideshow. Avoid any problems by using a value of 1 or greater.

Enabling Site-Relative URLs

Important: This topic describes an advanced technique. If you are not familiar with site-relative and document-relative URLs, or don't understand the difference between them, this topic isn't for you.

It is generally recommended that you always use the default setting of document-relative URL paths in your slideshows. Even in CMS environments, site-relative paths can usually be avoided. However, in certain instances you may have no other option than to use site-relative paths. Using site-relative paths has the benefit that the holding page can be moved from folder to folder without breaking the file references. However, it also means that **the built-in preview will not be available**; to preview, you will have to exit the wizard and preview in a browser.

To enable site-relative URLs for a slideshow, launch the Insert Slideshow wizard as normal. Before doing anything else, go to the Global Configuration / Playback Settings main tab and expand the Custom HTML Parameter pod. In the *Param Name* field enter `useSiteRelativeURLs` and in the *Value* field enter `true`:

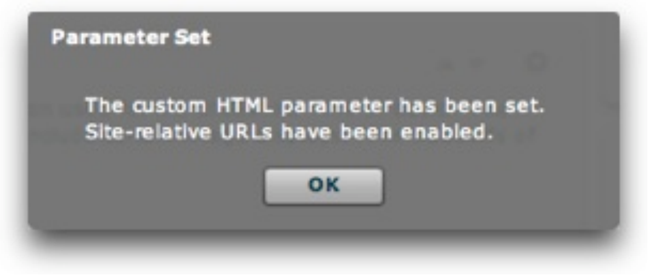


Custom HTML Parameters

These advanced options should only be used if you are familiar with what they control, or have been directed by suppport to do so.

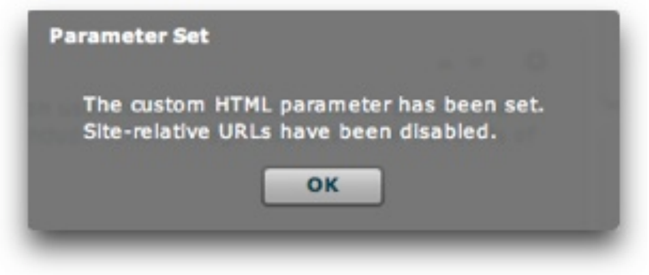
Param name: useSiteRelativeURLs Value: true

Click the Set Parameter button to apply the setting. You should see the following notification:


Parameter Set

The custom HTML parameter has been set.
Site-relative URLs have been enabled.

If you later wish to disable site-relative URLs and return to document-relative URLs, set the parameter again using a value of `false`. When you apply the parameter, you should see a corresponding notification:


Parameter Set

The custom HTML parameter has been set.
Site-relative URLs have been disabled.

Note: Disabling site-relative URLs on an existing slideshow will not modify any existing image paths; you must modify the paths manually as needed. Likewise, enabling site-relative URLs for an existing slideshow that already has images added will not affect the URLs of those images; their paths must be manually updated.

FAQ's / Common Issues

This section of the manual provides solutions to common questions and issues.   Each issues has a separate subtopic.

Suggestions for topics are welcome.

Do not use the comments below for support issues. If you need support, visit the forums or contact support via the Support Center.

My Slideshow Doesn't Show Up!

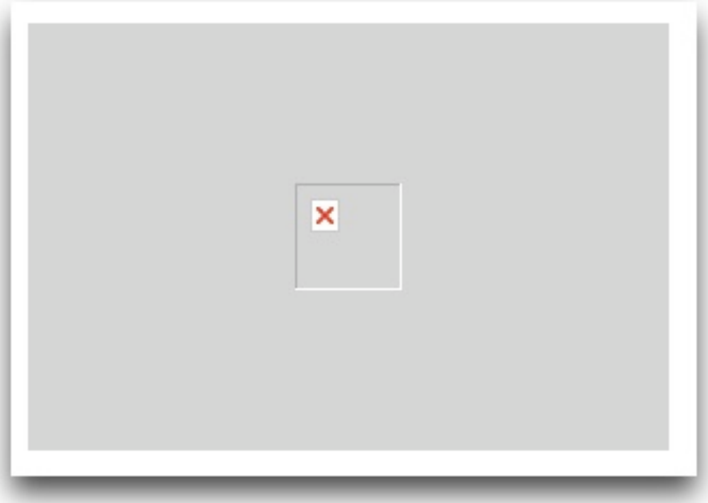
There are a number of common issues which can cause a slideshow to not appear. First, where are previewing the slideshow?

Previewing in the Wizard

If you're having trouble previewing the slideshow in the wizard, you are likely experiencing a Flash Player local security restriction. To learn about addressing these issues, see this topic.

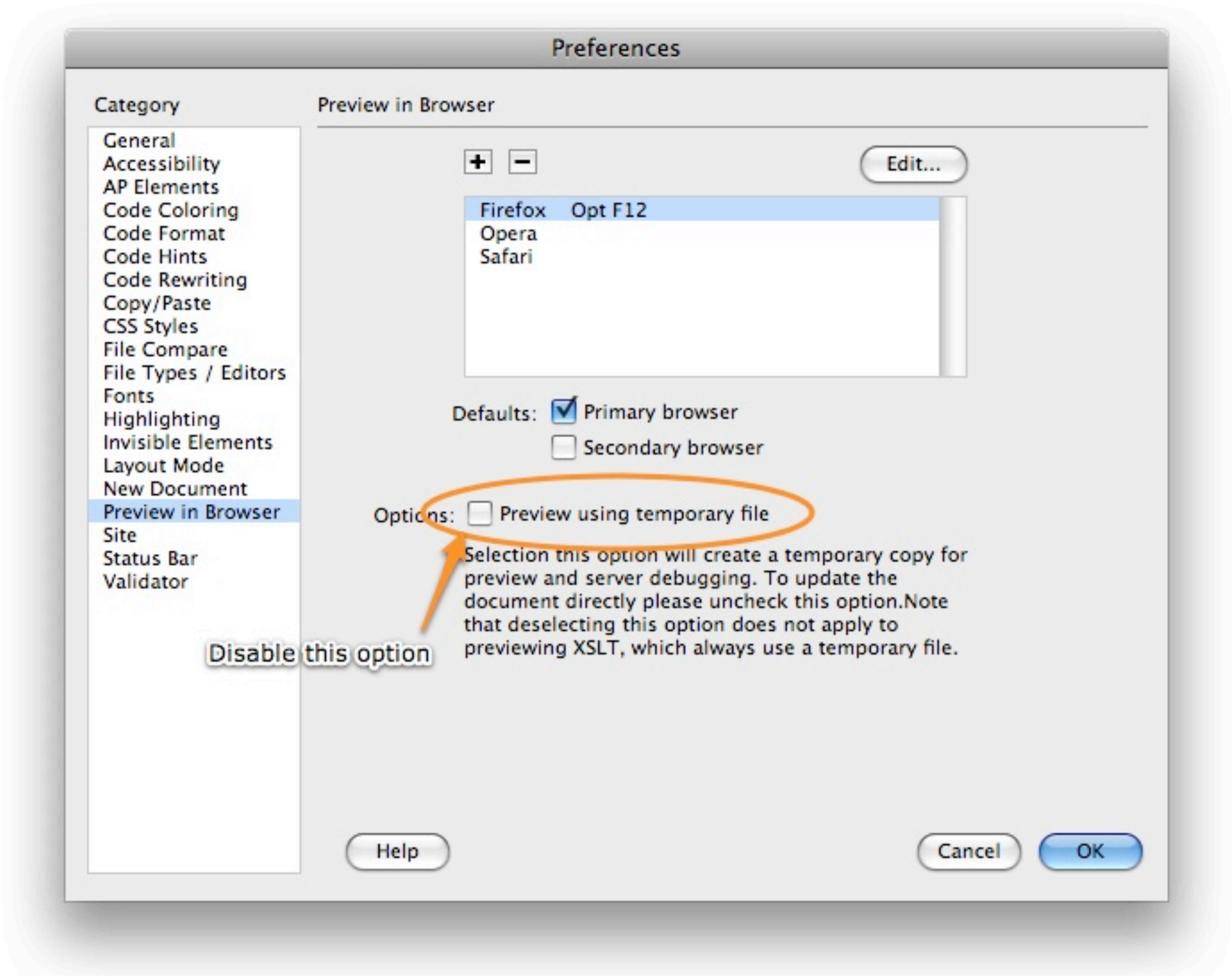
Previewing Locally (Using Dreamweaver's Preview In Browser Feature)

If you are previewing using Dreamweaver's Preview in Browser button, you may see a blank gray box with a broken image icon in the center:



To fix this, you must change a setting in the Preview in Browser section of the Dreamweaver Preferences dialog. To access the Preferences dialog, select Edit > Preferences... (Windows) or Dreamweaver > Preferences... (Mac). In the dialog, select the Preview in Browser section on the left and **disable** the Preview

Using Temporary File option:



Defaults: ☑ Primary browser

□ Secondary browser

Options: □ Preview using temporary file

Selection this option will create a temporary copy for preview and server debugging. To update the document directly please uncheck this option.Note that deselecting this option does not apply to previewing XSLT, which always use a temporary file.

Disable this option

Viewing On a Remote Server (Via an http:// URL)

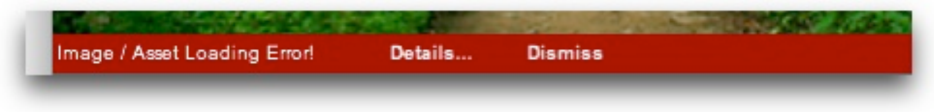
If you are trying to view the slideshow on a remote server and the slideshow doesn't appear or doesn't appear properly, you likely haven't uploaded all of the necessary files. The most important files are located in the `v4flashslideshow/` supporting files folder; the image files must be uploaded as well. To learn more about all of the files which need to be uploaded, see the Deploying Slideshows topic.

Previewing in Browser - Slideshow Only Doesn't Appear in IE

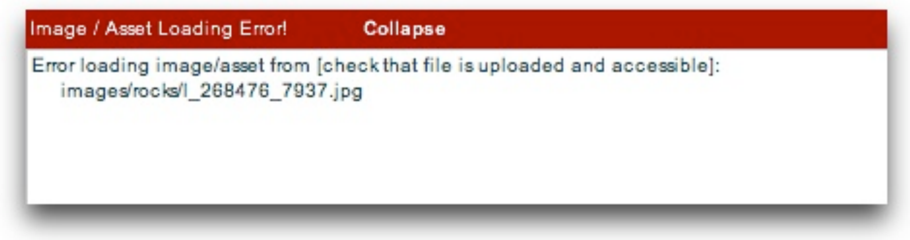
If the slideshow appears properly in all browsers except Internet Explorer, you likely have an IE ActiveX-fix Javascript script added to the page. These scripts interfere with the slideshow code and must be removed.

An 'Asset Loading Error' Message Bar Appears

When you view a slideshow, you may see the following bar appear at the bottom of the slideshow:



Additionally, the image(s) which failed to load will appear in the slideshow as a large 'Unable to Load Image (IO)' image. The error bar provides information to help you determine the problem and address it. Clicking 'Dismiss' will hide the bar entirely until another error occurs. Clicking 'Details...' will expand the window and show troubleshooting information:



Error loading image/asset from [check that file is uploaded and accessible]:
images/rocks/l_268476_7937.jpg

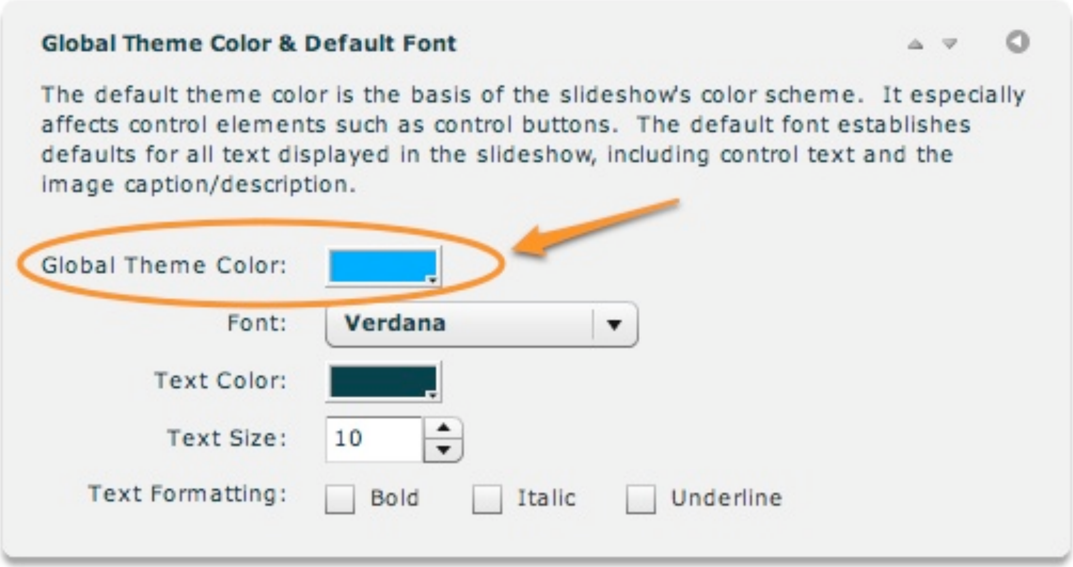
For each error, the box displays the image's URL and the problem which occurred (e.g. failed to load, or security restriction). **The most common source of image loading errors is failing to upload the image files.** For images to appear in the slideshow, they must be uploaded and accessible to be loaded at runtime.

You can further pinpoint the issue by copying the listed image URL and pasting it into your browser's address bar relative to the current directory. For the example above, if your address bar contained `http://www.example.org/` , you would paste the URL directly on so that it is `http://www.example.org/images/rocks/l_268476_7937.jpg` . If your address bar contained `http://www.example.org/somedirectory/somepage.html` , you would paste the URL over the filename so that it is `http://www.example.org/somedirectory/images/rocks/l_268476_7937.jpg` . When you load the URL, you will likely see a 404 Not Found error. You might also see a 403 Forbidden error; in these cases, check your server's file permissions.

How to Change Control Button Colors

Tip: Want to use entirely custom buttons? See the Custom Buttons topic for directions.

You've created your slideshow, but the default control buttons are a shade of blue which clashes with the site design. Fortunately, changing the theme color of all buttons is simple. Select the main Appearance Settings tab and open the Global Theme Color & Default Font pod. In this pod, change the Global Theme Color setting to some other color:



Global Theme Color & Default Font

The default theme color is the basis of the slideshow's color scheme. It especially affects control elements such as control buttons. The default font establishes defaults for all text displayed in the slideshow, including control text and the image caption/description.

Global Theme Color:

Font: Verdana

Text Color:

Text Size: 10

Text Formatting: □ Bold □ Italic □ Underline

The change will be reflected on all control buttons which don't have a theme color explicitly declared, as well as control elements such as the Thumbnails element's scrollbar.