Zoomify

Zoomifyer for *Flash*™

User's Guide

Box 8008 • Santa Cruz • CA • 95061

Table of Contents

Chapter I. Welcome to Zoomify

Zoomify unleashes unlimited resolution

Zoomify makes high-resolution images useful on the Web. The Zoomifyer family of products quickly converts images of any size or quality to *stream* for fast initial display and on-demand viewing of fine details!

Zoomify enables publishing of *gigabyte* photos that can be viewed without any large download. Visitors to your Web site can interactively zoom-in and explore huge images – of ultra-high quality – in real-time.

Zoomify's solutions are geared toward the needs of your Web site.

- Zoomifyer *EZ* enables your site visitors to interactively pan and zoom to explore images in detail with only Flash Player 6 or later needed – all for free!

- Zoomifyer for *Flash* enables your site visitors to interactively pan and zoom to explore images in detail with only Flash Player 6 or later needed. Or use ActionScript to dynamically drive the display to any view at any resolution.

- Zoomifyer *Enterprise* delivers mission-critical functionality essential to microscopy and other high-resolution imaging applications involving multi-gigabyte images as well as integrated implementations requiring server-side automation.

And Zoomify's solutions never involve any license fees, 'broadcast keys', or arcane contracts. In short, there are no strings on your images whatsoever. Simply convert and host on any Web server – anytime, anywhere.

How Zoomify's technology works

How does the Zoomifyer work? The Zoomifyer converter intelligently converts any image for *incremental* access. Macromedia Flash Player[1] is then able to display any view of the image – close or far, left or right, up or down – without delivering any unneeded, undisplayed image data.

The Zoomifyer begins by copying an image several times at different resolution levels – from the original source resolution down to a thumbnail. Zoomifyer cuts each of these levels into many small tiles. The tile concept is *pyramidal* – that is, like a pyramid, stacked from a thumbnail down to the highest resolution, level upon level. In the Zoomifyer converter, these same tiles are saved as separate images and organized into folders.

[1] Flash Player 6 and above.

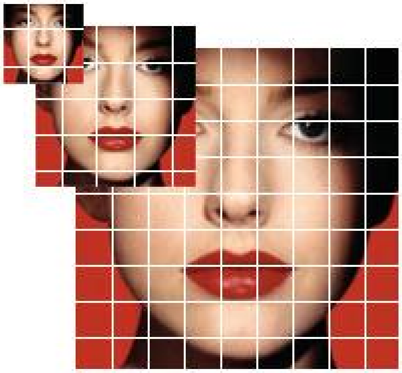


Figure 1: Pyramidal Tiled Multi-Resolution Image

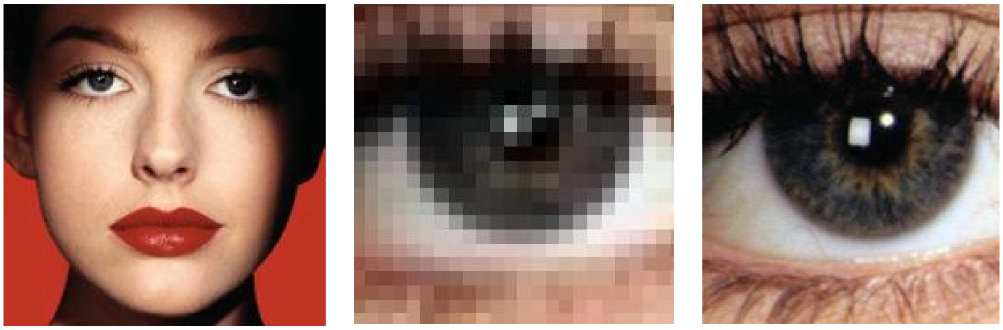
When the new file is viewed, the Zoomify Flash Viewer requests only the tiles necessary to display the current view from the Web server. Each pan and zoom causes a request for only a small additional number of tiles – those for the part of the image panned to, at the level of zoom desired. No tiles are ever delivered unless required for the current display, or for a display that is anticipated to immediately follow (intelligent *pre-fetching*).

These requests for image data are all made via standard HTTP Internet protocol. The only difference is that the Web server is providing *parts* of image files rather than *entire* image files.

Zoomify and compression

Zoomify's image streaming approach is complementary to compression because compression reduces the size of data delivered while Zoomify ensures that no unnecessary data is delivered. In fact, Zoomify leverages the complementary benefits of compression by applying JPEG compression to the image tiles within a converted file.

Compression alone is inadequate for publishing high-quality images. A 32 megabyte image licensed from a stock photo house – even if compressed 100-fold with wavelet technology to approximately 320k – is impractical for Web viewing. The time required to download a file of this size – or one much larger – makes it unreasonable to attempt to publish a 100-megabyte file without image streaming. Zoomify supports viewing of images *gigabytes* in size, which are difficult to view without Zoomify even if the file is on your local machine.



Example Image Enlarged Low-Res Image Zoomify Image

Figure 2: Image Streaming Illustrations

Zoomifyer delivers one-step image prep

Zoomifyer for *Flash* enables creative professionals to convert images to stream directly on the desktop with the Zoomifyer converter. Zoomify conversion presents a single, end-of-workflow step to prepare content for publishing on the Web. Simply double-click the appropriate Zoomifyer icon or drag an image onto it, and the Zoomifyer does the rest. To convert multiple images at once (batch conversion) simply drag multiple images onto the icon, or if double-clicking the application, select multiple files in the File Open dialog by shift-clicking.

Using this Guide

This Guide describes the features of Zoomifyer for *Flash*. Users new to imaging, the Web, Zoomify, or Macromedia Flash will most likely prefer to follow the chapters in the order presented. The section entitled “Zoomifyer for *Flash* Basics” can be read now or referred to once you’ve become more familiar with the Zoomifyer and its capabilities.

Contacting Zoomify Support

For additional help Zoomify’s outstanding Support Team is standing by. Simply email *support@zoomify.com* - every effort is made to respond promptly. Please be sure to first review relevant portions of this Guide prior to contacting us. Standard business hours are 8 a.m. to 5 p.m. Pacific Time. Don’t forget to check *http://www.zoomify.com* regularly for the latest Zoomify news, viewer updates, and more. For additional help, please be sure to visit our support pages at *http://www.zoomify.com/support/*.

If reporting a problem with the product, please provide the following information to the best of your ability:

Description:	Area of product affected. Application, browser, and operating system behavior.

Errors: Exact wording of any error message displayed.
Steps to reproduce: Exact order of actions leading up to the problem including every menu and button choice.
Reproducibility: Is the problem 100% repeatable or intermittent?
Configuration: Platform and model, operating system and version, browser and version, and Zoomifyer product and version.

We are committed to your successful use of Zoomifyer for *Flash*. Thank you for your business.

The Zoomify Team

Chapter II. Zoomifyer for *Flash* Basics

If you're already familiar with Zoomify's product line or just want to jump right in, this section is for you! This step-by-step outline will explain how to install Zoomifyer for *Flash*, create a Zoomify image, and view your new zoomable image – all within the course of a few minutes.

For the simplest example of how to use Zoomifyer for *Flash*, please see the file called "Quick Start.htm" included with the product – it's located at the root of the Zoomifyer for *Flash* folder. In order to integrate Zoomifyer for *Flash* into Flash MX or MX 2004 please follow the steps below.

Installing Zoomifyer for **Flash**

First, please verify that you've installed and run Macromedia Flash MX or MX 2004 at least once. This will ensure that the files and folder structure required by Zoomifyer for *Flash* have already been put in place.

If you've purchased Zoomifyer for *Flash* online, you'll download a ZIP or SIT file to your computer. On the Mac your browser will likely extract the SIT file automatically. If not, you can obtain a free version of a program called "StuffIt Expander" from *http://www.aladdinsys.com*. On windows you'll need a program called "WinZip". A free trial is available at *http://www.winzip.com*. Once you've decompressed the files from your ZIP or SIT you'll have a folder entitled "Zoomifyer for *Flash*".

After ensuring that Macromedia Flash MX or MX 2004 is not currently running, open the Zoomifyer for *Flash* folder and double-click the installer called "zFlash v3 MX Installer" or "zFlash v3 MX 2004 Installer", depending on which version of Macromedia Flash you own. You will see a splash screen with "Install Later" and "Continue" buttons. Just click "Continue" to install Zoomifyer for *Flash* on your system.[2]

Creating a new Zoomify project

Once the installer has indicated that it has completed the installation process, run Flash MX or MX 2004 and create a new Flash document (from the menu: *File* > *New*). To access the Zoomify Components, you may need to either expand the Flash Components panel or, if it is not present in your panel set, open it from the menu. In Flash MX, open the menu item *Window* > *Components*. In Flash MX 2004, open the menu item *Window* > *Development Panels* > *Components*.

Once you have access to the Zoomify Components in the Flash Components panel, drag the component called "ZoomifyViewer" onto your stage. Then select the component with

[2] If you have any trouble with the installer, please see the appendix entitled "Manual Installation of Zoomifyer for *Flash*" for instructions on how to manually install the Zoomifyer for *Flash* files.

your cursor and, in the Properties panel, give it an instance name such as "test". This instance name is important as it allows you to link other Zoomify Components to the ZoomifyViewer component. Once you've given your ZoomifyViewer component an instance name, drag on the "ToolbarStandard" component and select it. In the Properties panel type "test"[3] in the "Target ZoomifyViewer" field. Then, save your project out as a FLA file.

Create a Zoomify Image

Now, click on the ZoomifyViewer component you previously placed on the stage. In the Properties panel you'll see a button entitled "Launch Component Inspector". Click this to launch the ZoomifyViewer Component Inspector and create an image to associate with your project.

Note that the first time you launch the ZoomifyViewer Component Inspector it will likely appear with other panels such as the Components panel or the Color Mixer panel. You may have to drag it out onto the stage and then enlarge it before using it.

In Flash MX 2004, click the "Convert" button next to the "Image Folder Path" text field. You'll then be prompted to select an image via a standard browse dialog. If you're using Flash MX, you'll need to drag an image onto the "Zoomify Converter" application at the root of the Zoomifyer for *Flash* download package – or simply double-click the Converter application to select an image.

For best results when encoding images, we recommend that you use a bitmap (.bmp), TIFF (.tif or .tiff), PNG (.png), or other losslessly compressed image. The Zoomify converter applies JPEG compression. In order to obtain the best possible quality you'll want to begin with a source image that has not been previously compressed with a 'lossy' image format.[4] Once you've selected an image to convert, you'll then be prompted to save the Zoomify Image folder that will be created during the conversion process. For now we recommend you save it in the same directory that you've already saved your project file (FLA file). You should then see the image convert in a popup window. Once the conversion is complete, the popup window will close.

If you are converting from within Flash MX 2004, you should click the "Refresh" button to see the image appear within the Component Inspector window. For Flash MX, you'll need to enter the path to the Zoomify Image folder. Although you can't use standard keyboard shortcuts to copy and paste into the Component Inspector, you can right-click in the text field (control-click on the Mac).

[3] Or whatever instance name you've given to your ZoomifyViewer component.

[4] Zoomify offers native support for BMPs (Windows bitmap files), TIFFs, and JPEGs. If you have QuickTime 5+ installed, you can convert any image format that QuickTime supports. Please be aware that native conversion tends to be faster, and you might wish to convert your images to the bitmap or TIFF format before Zoomifying them.

View your new Zoomify Image

You can now interact with your Zoomify project. You can first hide the Component Inspector by minimizing it or closing it. Then, select the menu choice "*Control > Test Movie*" to run the movie and present the newly created image, complete with Toolbar, for easy testing of zoom and pan interactivity![5]

Copy to your Web server

To publish your Flash project, select the menu choices "*File > Publish*" to create the SWF and HTML files. Then, simply copy the SWF, the HTML file, and the Zoomify Image folder you created to your Web server. That's it!

Of course, the preceding is the simplest implementation of Zoomifyer for *Flash* – it's completely customizable, extensible, and compatible with your pre-existing Flash MX or MX 2004 movies.

[5] If you have any trouble creating or viewing you newly Zoomify Image, please see the Chapter entitled "Using Zoomifyer for *Flash*".

Chapter III. Elements of Zoomifyer for *Flash*

Quick Start.htm

Read this first for information on how to rapidly create and deploy Zoomify content.

Zoomify Converter application

Preparing images for interactive zoom and pan using the Zoomify Components can be accomplished right within Macromedia Flash MX 2004. As noted above in Zoomifyer for *Flash* Basics, the conversion capability can be accessed seamlessly at the time the ZoomifyViewer component settings are specified in the Component Inspector (custom properties panel). The Zoomify Command is also provided on the Macromedia Flash MX 2004 Commands menu. For desktop conversion outside Macromedia Flash MX 2004 or for Flash MX users, a standalone converter is also provided. The Zoomifyer for *Flash* image standalone converter is implemented as a 'droplet' – an extremely simple tool upon which a source image is dropped to initiate conversion. Alternatively, double-clicking the droplet will present a standard "file open" dialog. Zoomifyer for *Flash* is so simple it actually requires no interface other than this (optional) open dialog and the progress display presented once conversion begins.

Zoomifyer for Flash *Installer*

The installer copies the files necessary to integrate Zoomifyer for *Flash* with Macromedia Flash MX or MX 2004. Just double-click the appropriate installer to start the installation process ("zFlash v3 MX Installer" for Flash MX and "zFlash v3 MX 2004 Installer" for MX 2004).

Documentation folder

a. Legal Notices.txt

   Terms of use for Zoomify products. Developers, please read the Legal Notices for appropriate use of Zoomify's source code.

b. Register-Support-Upgrade.htm

   For information on registering your copy of Zoomifyer for *Flash*, obtaining technical support, and learning more about upgrade products, please double-click this file to open it into your browser. This file can also be accessed using the Help button when viewing the Component Inspector for the ZoomifyViewer Component (custom properties panel).

c. ZF User's Guide v3.0.pdf

   The Documentation folder contains the Zoomifyer for *Flash* User's Guide that provides information on working with Zoomifyer for *Flash*.[6]

d. Sample Content *folder*

   Contains a link to download sample content from Zoomify's site, a sample JPEG file you can use to test the Zoomifyer for *Flash* converter (sampleImageToConvert.jpg), and a sample Zoomify project FLA (SampleProject.fla) that can be used to quickly create a Zoomify image project.

e. Examples *folder*

   A collection of developer examples showing how the ZoomifyViewer Component may be modified, including a custom toolbar implementation, an example of smoothly zooming to a specified view, and more.

Designer Tools **folder**

The Zoomify Flash Component is designed to rapidly stream relevant views of very large high-quality images over the Web. You can also view the files locally by entering the name of your folder into "ZoomifyDynamicViewer.swf".

a. Simple Web Page *folder*

   Includes the files "zoomifyDynamicViewer.htm" and "zoomifyDynamicViewer.swf", which can be immediately used without any editing in Macromedia Flash MX or MX 2004 to enable interactive zoom and pan of any Zoomify image. The Dynamic Viewer supports numerous HTML parameters that can be set to specify what image to display and what view of that image to initially present, as well as to control the look and feel of many aspects of the Viewer. Also includes a short explanatory text file.

b. URL-Driven Web Page *folder*

   Includes the file "zoomifyURLDrivenWebPage.htm" which can accept parameters (including Zoomify image paths, toolbar information, and more) via URL. Also includes a short explanatory text file.

Developer Tools **folder**

[6] Published in Adobe's Acrobat format – the reader is free from *http://www.adobe.com/*.

The Developer Tools folder contains files automatically installed by Zoomifyer for *Flash* that are typically accessed from the Macromedia Flash MX 2004 menus. See the following section titled "Zoomifyer for *Flash* Integrated Features" for more information..

a. Behaviors *folder*

   The Zoomify Behavior enables you to add Zoomify functionality[7] to pre-existing Flash MX 2004 symbols and components. Not available in Flash MX – use the Zoomify API functions instead.

b. Commands *folder*

   Allows access to the "Zoomify Converter" application from within Flash MX 2004. Not available in Flash MX – use the included Zoomify Converter application instead.

c. Custom Actions *folder*

   The Zoomify Custom Actions provide direct access to descriptions of every API call supported by the Zoomify Viewer Component. For additional information see the section below titled "Zoomifyer for *Flash* Integrated Features."

d. Installer Support Files *folder*

   Files used by the Zoomifyer for *Flash* installer.

e. Zoomify API Reference *folder*

   The Zoomify API Reference provides direct access to detailed descriptions of every Zoomify ActionScript API call supported by the Zoomify Components.

f. Zoomify Components *folder*

   Contains the "Zoomify Components.fla" and "Zoomify Components 2.fla" files that are integrated into Flash MX and MX 2004 by the Zoomifyer for *Flash* installer.

[7] Such as "zoomIn", "zoomOut", "panLeft", etc.

Chapter IV. Using Zoomifyer for *Flash*

Zoomifyer for Flash *Integrated Features*

Once you've successfully installed Zoomifyer for *Flash*, you can access its features, the helps files, and Zoomify Behavior functions from within Flash MX or MX 2004. If you've had difficulty installing Zoomifyer for *Flash* into your copy of Macromedia Flash MX or MX 2004, please see the section below entitled "Troubleshooting installation" which should help you resolve any issues you may encounter.

Zoomify Components

The heart of the Zoomifyer for *Flash* is the file "Zoomifyer Components.fla", access to which is integrated directly within Flash MX or MX 2004. It contains the following components:

1. *ZoomifyViewer*

   Every Zoomify project requires a ZoomifyViewer component. It's simple to use – just open the Flash MX or MX 2004 "Components" panel (from the "*Window > Development Panels*" menu item, expand it if necessary, and then drag the ZoomifyViewer component onto the stage.

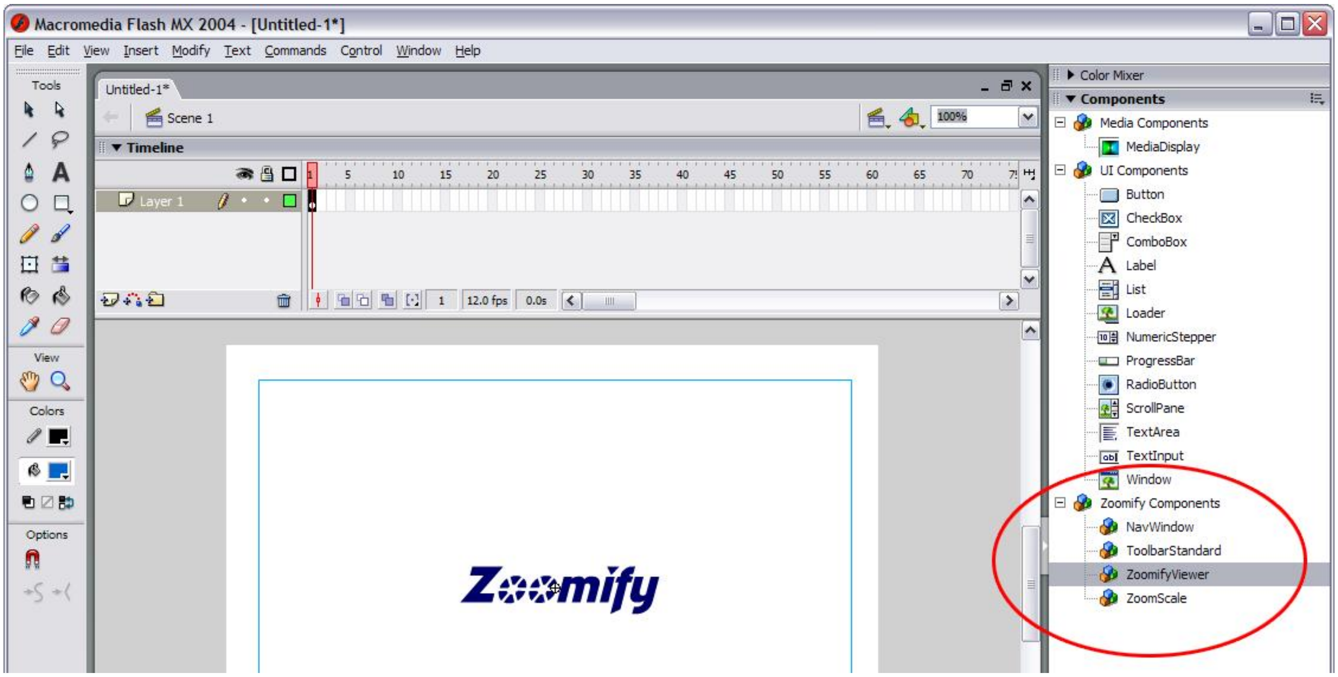
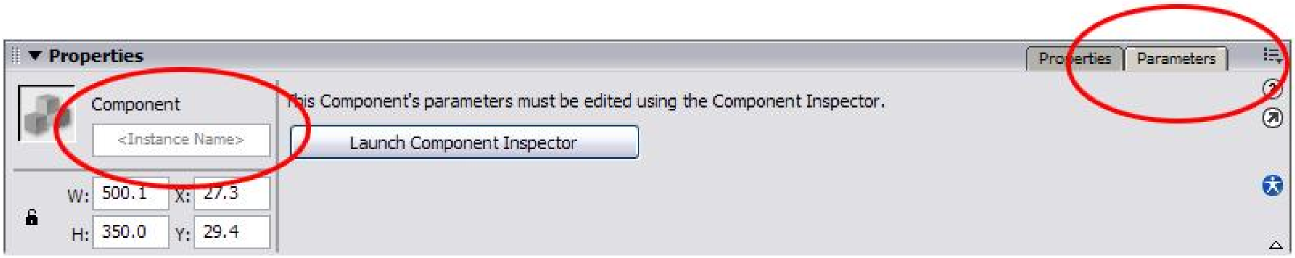


Figure 3: The Zoomify Components in Flash MX 2004

The first thing you'll want to do is give your new ZoomifyViewer an instance name so it can be identified and linked to other Zoomify components such as the ToolbarStandard component. The instance name section is located on the Properties panel on the Parameters tab.



This Component's parameters must be edited using the Component Inspector.

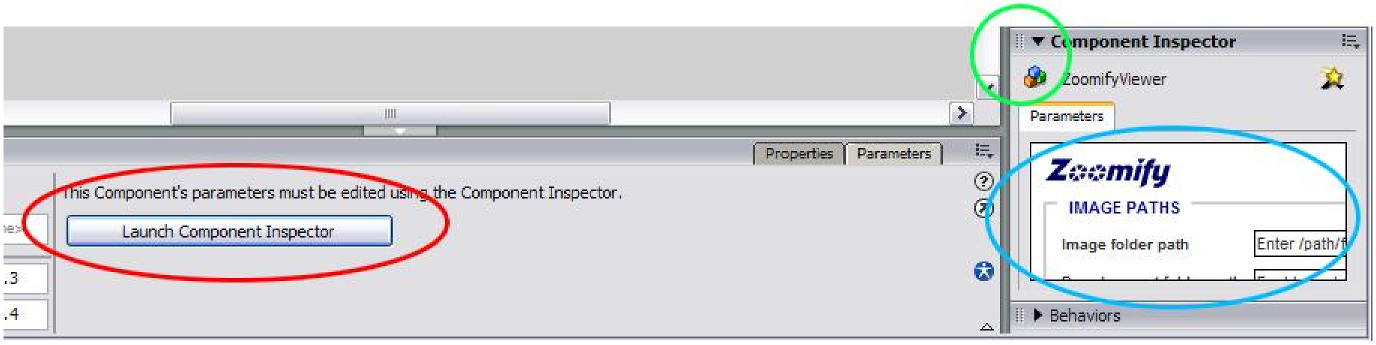
W: 500.1 X: 27.3
H: 350.0 Y: 29.4

Figure 4: Giving the ZoomifyViewer component and instance name.

You can also change the ZoomifyViewer component's size by clicking in the width and height text fields beneath the instance name.

Once you've done this, you should save your new Zoomify project as a FLA file.

Then, from the Properties panel, click the "Launch Component Inspector" button to launch the ZoomifyViewer Component Inspector.



This Component's parameters must be edited using the Component Inspector.

Figure 5: Launching the Component Inspector

Please note that when you first launch the ZoomifyViewer Component Inspector, it will likely appear with the panels on the right-hand side of Flash MX or MX 2004 (see blue highlighting above). You'll need to use the drag handle (green highlighting) and drag the Component Inspector onto the stage to undock it from the right-hand side panels and move it to the center of your screen. From there, you can enlarge the Component Inspector window until you can see it completely.

The ZoomifyViewer Component Inspector is where you'll actually convert (or designate) the Zoomify Image that will appear in your project, modify some properties about that image (such as its original X and Y coordinates and zoom level), and edit some properties about the ZoomifyViewer itself (such as whether the splash screen should be displayed, etc.)

If you haven't already created a Zoomify Image Folder to work from (by using the image conversion method detailed below), you should then click the "Convert" button to select an image to convert (MX 2004 only). You'll be presented with a standard browse dialog in order to select an image, and also be prompted for a location to save your new Zoomify Image.

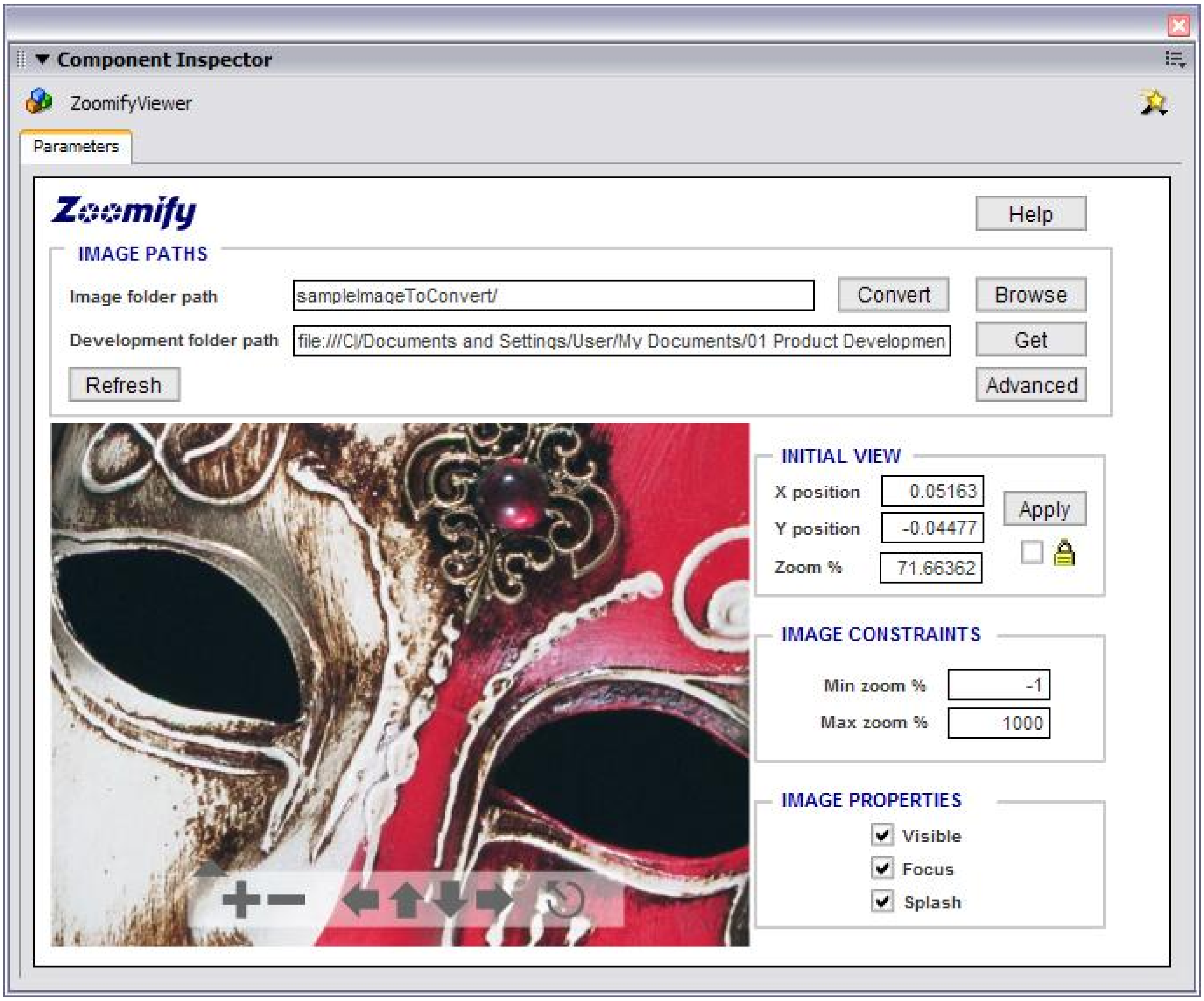


IMAGE PATHS

Image folder path: sampleImageToConvert/

Development folder path: file:///C|/Documents and Settings/User/My Documents/01 Product Developmen

INITIAL VIEW

X position: 0.05163

Y position: -0.04477

Zoom %: 71.66362

□

IMAGE CONSTRAINTS

Min zoom %: -1

Max zoom %: 1000

IMAGE PROPERTIES

☑ Visible

☑ Focus

☑ Splash

Figure 6: The ZoomifyViewer Component Inspector

Please note: more detailed information on image conversion is available below in the section entitled "Alternative Image Conversion: drag & drop", including the best file formats to use, how to batch convert, and more.

Once you've successfully selected an image and its destination directory, the image will convert in a small popup window. When it's done, click the "Refresh" button to display the new Zoomify Image in your Component Inspector.

You can also set some parameters about the Zoomify Image itself. In the "Initial View" section, you can set the X, Y, and zoom levels of the image's initial appearance. You can use the toolbar in the view area to navigate to the view you'd like to display when the Zoomify Image is first viewed. The "Lock" checkbox allows you to save desired settings while still navigating the image. To move the image back to the locked settings, simply click "Apply".

The "Image Constraints" section allows you to set zoom settings for the Zoomify Image. In the "Min zoom %" field the default value is "-1" (recommended) which indicates that the image cannot be zoomed out beyond the borders of the display area. A setting of "1" or less (e.g., .01, .0001, etc.) will allow the image to be zoomed out until is appears much smaller than the display area. The "Max zoom %" field refers to the pixel depth of the image itself. A setting of "100" (recommended) will allow the image to be zoomed to the pixel depth of the original source image. A setting below 100 will constrain the zoom on a percentage basis. To "over-zoom" the image, settings over 100 may be used. With certain content, settings of 120, 150, or even 200 can be used to provide greater zoom without noticeable image quality degradation.[8]

The "Image Properties" section allows you to set some properties about the ZoomifyViewer component that will be displaying the Zoomify Image. The "Visible" checkbox determines whether or not the ZoomifyViewer component instance will appear when the SWF is first loaded. This is checked by default. The "Focus" checkbox determines whether or not the ZoomifyViewer component instance will receive keyboard events. The "Splash" checkbox determines whether or not the splash screen (by default, the Zoomify logo) will be displayed when the SWF is first loaded. (The splash screen is completely customizable – just swap the Zoomify logo graphic in the library).

2. *ToolbarStandard*

Another important Zoomify component is the "ToolbarStandard" component. This component allows the end-user to interact with the ZoomifyViewer component in an intuitive fashion.

The ToolbarStandard can be dragged onto the Flash MX or MX 2004 stage in the same way as the ZoomifyViewer component. Once it's on the stage, you'll have to associate it with a ZoomifyViewer component by typing in the appropriate ZoomifyViewer component instance name into the "Target ZoomifyViewer" text field in the Properties panel. You can also set the "Show Slider" parameter to true or false.

3. *NavWindow*

The "NavWindow" component presents a "bird's eye" view of the Zoomify image. Whereas the main view window image (the ZoomifyViewer component area) is zoomed and panned, the NavWindow image is static and is designed to aid navigation within the larger image. It does have a rectangle indicating the current view within the zoomable image itself, and this rectangle may be moved to change the main view.

[8] Note however, that such 'false zoom' is not a substitute for a higher quality source image. The strength of Zoomify is providing fast, interactive access to all the quality of your image no matter its size or resolution.

Again, you'll need to enter the appropriate ZoomifyViewer component instance name into the "Target ZoomifyViewer" text field in the Properties panel. You may also enter the paths to the Zoomify image and the development folder path (identical to the ZoomifyViewer Component Inspector paths) if a preview is desired. These paths are completely optional and if left blank the NavWindow component will inherit the appropriate image data from the targeted ZoomifyViewer once the Zoomify Flash project is published or the movie is tested (*Control* > *Test Movie*).

4. *ZoomScale*

The "ZoomScale" component is quite useful for more technical image data such as medical slide scans or satellite/aerial photographs. By specifying some basic measurement data about the image in question, you'll enable the ruler, which can then display scale information about the image at any zoom level.

After dragging the ZoomScale component onto the stage and setting the "Target ZoomifyViewer" text field to match with the ZoomifyViewer component in question, you'll then need to set some other values.

- PixelsPerUnit: if known, enter this value (in lieu of UnitsPerImage). For example, if you're working with a satellite photo, you might have a one-to-one correspondence between pixels and meters, hence 1 pixel per unit.
- UnitsPerImage: if known, enter this value (in lieu of PixelsPerUnit). For example, if you know the image is 1000 meters across, you would enter 1000 here.
- Source Magnification: if known, enter this value. Used in conjunction with PixelsPerUnit.
- Ruler Units: select the unit of measurement that corresponds with your data (i.e., m, cm, mm, etc.)

Behavior functions

In Flash MX 2004, you can assign specific Zoomify behaviors to pre-existing components, symbols, and the like. The simplest example of this would be to drop a Flash MX 2004 component button on the stage and assign it "zoomIn" or "zoomOut" functionality such that it can zoom in and out of a Zoomify Image.

To work with Zoomify Behaviors, you may need to open the "Behaviors" panel by clicking the "*Windows* > *Development Panels*" menu and selecting "Behaviors".

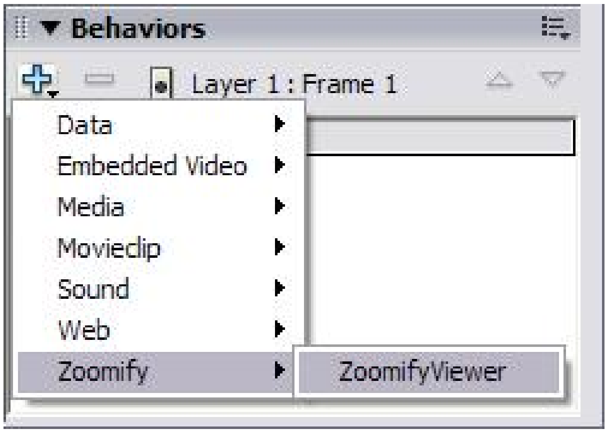


Figure 7: Flash MX 2004 Behaviors Panel

Simply select the object you wish to assign the behavior to (such as a button component) and select the ZoomifyViewer behavior from the list. Then select a ZoomifyViewer component to control, and then assign the behavior function itself.

Currently, the list of possible behavior functions includes:

1. zoomIn
2. zoomOut
3. panLeft
4. panRight
5. panUp
6. panDown
7. resetView

In the Behaviors panel, you can also assign the Event that triggers the function, such as "Click" for the aforementioned button.

Zoomify Converter Command

You can call the Zoomify Converter from within Flash MX 2004 in one of two ways. First, when you've selected a ZoomifyViewer component and launched its component inspector, you can simply click the "Convert" button to select an image to convert. Second, you can call the Zoomify Converter at any time by selecting "*Commands > Zoomify Converter*" from the menu bar.

Also, you can batch convert multiple Zoomify images by simply selecting multiple images in the file select dialog box.

For more information on converting images into Zoomify Image folders, please see the section below entitled "Alternative Image Conversion: drag-and-drop".

Zoomify API Reference / Custom Actions

The Zoomifyer for *Flash* installer will also install a Zoomify API Reference "book" within the Flash MX 2004 ActionScript Dictionary. To view the Zoomifyer API Reference, simply select the "*Help > ActionScript Dictionary*" menu item and locate the "Zoomify API Reference" book.

The Zoomify Custom Actions provide code hints for the Zoomify API within the Macromedia Flash MX 2004 editor. To use the Zoomify Custom Actions, select the "*Window > Development Panels > Actions*" menu item.

In Flash MX, you can access the Zoomify API Reference from "*Window > Reference*" and selecting "Zoomify Components".

Alternative Zoomify Image conversion: drag & drop

To convert images for use with Flash MX, (and outside Flash MX 2004 if you prefer), you can use the application entitled "Zoomify Converter" located at the root of the Zoomifyer for *Flash* product folder.

To convert an image, simply drag it onto the "Zoomify Converter" application target icon – the conversion window will pop up and you can watch the conversion process as it happens. You may also double-click the converter and select an image from a dialog box. Once the conversion completes, you will have a folder with a name identical to your original image.

Because the new Zoomify Image folder will be created in the same directory as the original image, ensure that you have write privileges to the directory in question. If you wish to convert an image from a read-only directory or a CD-ROM, first copy the image to a folder with write permissions enabled. If you have previously converted an image with the same name in the same directory and wish to keep it, please rename the existing Zoomify Image folder or move it to another directory. Zoomifyer for *Flash* will overwrite folders with the same name.

The time it takes to convert an image depends chiefly on the size of the image and whether it is a format native to Zoomifyer for *Flash*. Other factors include your system specifications,[9] your operating system,[10] and the number of other applications or processes running on the machine.

You can also batch convert images by simply selecting multiple image files and dragging them onto the Zoomifyer for *Flash* icon. Zoomifyer for *Flash* will convert them sequentially. Please ensure that you have enough disk space for both the original images and the newly created Zoomify images. If you plan to convert a large number of images that are not

[9] The most important specifications are the processor speed, RAM, hard disk speed and space available.

[10] Generally speaking, encoding an image will take less time on machines running Windows 2000 or XP than under Windows 98. Similarly, conversions will generally take less time under Mac OS X than under Mac OS 9.

native to Zoomifyer for *Flash* via the batch process, you might consider running a batch process in your favorite image-editing application and saving out your original images as BMPs or TIFFs.

Zoomifyer for *Flash* supports BMP, TIFF, and JPEG files natively.[11] We recommend that you use a non-compressed image format such as BMP or TIFF when converting to the Zoomify image format. You may also save and convert JPEG and other 'lossy' compressed formats that offer very high compression ratios. It is important to note that the conversion process itself will apply JPEG compression to the final zoomable content. Beginning with a file that has already been compressed in a lossy manner will result in double JPEG compression. This may lead to a noticeable loss of image quality. As a general rule of thumb, Zoomify the best quality source image you have to achieve the best quality zoom.

If you have QuickTime 5 or higher installed on your machine, Zoomifyer for *Flash* can convert any file format supported in QuickTime, such as PICT, PNG, etc. QuickTime is available for free from *http://www.apple.com*. Please note that you do not need the "Pro" version of QuickTime to use the image libraries with Zoomifyer for *Flash*. Also, conversion with formats that rely on QuickTime format libraries will take more time than with native formats.

Publishing Your Zoomify Image

Once you've added Zoomify components to the stage of your Flash MX or MX 2004 movie, you can publish the Zoomify Flash movie by publishing it as you would any Flash MX 2004 movie, i.e., by selecting the "*File > Publish*" menu item. This will create a SWF file that includes references to your Zoomify Image. Now, simply copy the HTML file, the SWF file, and the Zoomify Image folder to the Web server, and that's it!

In addition to publishing a SWF file, there are two other very simple ways to publish a Zoomify Image.

Using the "Simple Web Page" (ZoomifyDynamicViewer.swf)

Zoomifyer for *Flash* includes a simple SWF file and an associated HTML file – just open the "Designer Tools | Simple Web Page" folder. Parameters can be easily edited to direct the SWF to display any Zoomify image. This approach is preferable for customers who prefer editing HTML to editing FLA parameter settings, as well as for customers who do not yet own Macromedia Flash MX or MX 2004. To make use of these 'dynamic' files, it is only necessary to edit the HTML to specify the path of the image folder to display.

[11] Native format support means that the file format decoder is built into the *Zoom*ifyer for *Flash* tool and is therefore directly utilized during the conversion process.

To edit the image folder this HTML and SWF refer to, you'll need to edit the HTML file. Open it with your favorite text editor. You'll see a standard Flash <object> and <embed> statement.

Change the "zoomifyImagePath" variable from

```
http://www.zoomify.com/content/test/
```

to the right path for your folder on your server. You'll need to make this change in two places in the HTML, once in the PARAM NAME section, once in the EMBED SRC section. In each case, the image path variable is preceded by the statement "zoomifyImagePath="

Note that numerous additional parameters can also be easily set. These include:

a. zoomifyImagePath

   The path to your Zoomify Image folder or Zoomify PFF file.

b. bgcolor

   The color of the viewer display area (when the Zoomify image being displayed does not fully cover the display area).

c. zoomifyMinZoom

   Controls how far out the image can be zoomed.

d. zoomifyMaxZoom

   Controls how far in the image can be zoomed – 100 is recommended for most applications (i.e., 100 = 100% of the original image pixels).

e. zoomifyNavWindow

   Displays the navigation tool window in the Zoomify Flash viewer. A setting of "1" will display the navigation window, a setting of "0" will hide the window. The navigation tool is off by default.

f. zoomifyToolbar

   Displays the Zoomify toolbar in the Zoomify Flash viewer. A setting of "1" will display the toolbar, a setting of "0" will hide the toolbar. If the toolbar is hidden, the user can still interact with the image by pressing the "A" key to zoom in and the "Z" key to zoom out. The toolbar is on by default.

g. zoomifyX

Sets the initial display of the image within the viewing area on the X axis.

h. zoomifyY

Sets the initial display of the image within the viewing area on the Y axis.

i. zoomifyZoom

Controls how far in your Zoomify image is zoomed when it is first displayed.

j. zoomifySlider

Displays the zoom slider button on the toolbar. A setting of "1" will display the slider, a setting of "0" will hide the slider. The slider is on by default.

k. zoomifyNavX

Sets the navigation window coordinates on the X axis. If not set, zoomifyNavX will default to "0" which is at the left-hand side of the viewing area. Expressed in pixels.

l. zoomifyNavY

Sets the navigation window coordinates on the Y axis. If not set, zoomifyNavY will default to "0" which is at the top of the viewing area. Expressed in pixels.

m. zoomifyNavWidth

Sets the navigation window width. If not set, zoomifyNavWidth will default to "130", expressed in pixels.

n. ZoomifyNavHeight

Sets the navigation window height. If not set, zoomifyNavHeight will default to "130", expressed in pixels.

The easiest way to set these parameters is to use the "Zoomify HTML Maker" HTML and SWF files. Even if you are familiar with Flash object and embed statements, the HTML Maker will allow you to display your image in precisely the way you like by providing the "zoomifyX", "zoomifyY", and "zoomifyZoom" values. The "Zoomify HTML Maker" can be downloaded from *http://www.zoomify.com/downloads/*.

Using the "URL-Driven Web Page" (zoomifyURLDrivenWebPage.htm)

The "URL-Driven Web Page", also in the Designer Tools folder, is an HTML file written in JavaScript that allows you to pass in, via URL, certain parameters (as listed above in the

"Simple Web Page" section) and the Zoomify Image path itself. The "URL-Driven Web Page" accepts the same parameters as the "Simple Web Page" above. See the text file included in the "URL-Driven Web Page" folder for more information.

How the Flash Player Handles HTML Parameters

Macromedia Flash Player provides standard, built-in support for parameter management. The Player automatically looks for an additional parameter labeled 'FlashVars' within the Flash embed statements. If this is present, the Player reads it when it loads. If the FlashVars parameter includes multiple values delimited by a '&' character, the Player automatically parses them. These values are then available to any ActionScript in the movie being played. The Zoomifyer Dynamic viewer is simply a thin wrapper around the Zoomifyer Viewer Component that includes several ActionScript statements that test the availability of parameter values and if present, passes them into the Zoomify Viewer Component for use.

Special Note for Developers

The 'wrapper' movie described above can be modified to handle additional parameters by editing the "ZoomifyDynamicViewer.fla" file in the Flash MX 2004 editor. The FLA file is included in the Dynamic Web Page example in "Documentation | Examples". (Note that this example offers the additional benefit of dynamically inserting the FlashVars parameter based on the web page URL - see the ReadMe file accompanying the example for additional information.)

More information about passing general Flash 6 parameters into components is available on the Macromedia Web site. Please visit:

http://www.macromedia.com/support/flash/ts/documents/flashvars.htm.

Resources

Developer Examples: The folder "Developer Tools | Examples" contains numerous examples showing how the Zoomify Viewer Component may be modified. To view the examples double-click the file "Start Here.htm" or open any of the example folders and double-click the individual example HTM files provided. The examples include:

1. Nav Window: A navigation window enables a user to track their current view within the context of an entire image. A more advanced example, the ActionScript provided both tracks the current view and enables navigation through direct manipulation of the Nav Window.

2. URL-Driven Web Page: For sites involving large numbers of images it is important to ensure that implementing Zoomifyer Viewer is not only easy, but involves the

fewest possible files. This reduces opportunities for error and makes future modifications simpler. The Zoomify Dynamic Web Page HTML file enables any Web site to display many images using only a single HTML file and a single Flash SWF file.

3. Custom Toolbar: Implementing a custom toolbar using the Zoomify Viewer component is quite simple. This example illustrates the necessary steps.

4. Smooth Transition: Simple but effective, this example demonstrates the use of the setView function along with an interval to guide a smooth transition from one view to another.

5. Hotspots: Hotspots are achieved in a Zoomify view through the use of the ZoomifyHotspot component or the Zoomify Label API. This example demonstrates the relevant API calls.

6. Slideshows: Zoomifyer for *Flash* now includes a ZoomifySlideshow component that allows the straightforward creation of slideshows with transitions. The ZoomifySlideshow component may also be used to create objects and panoramas.

HTML Maker: The Zoomify HTML Maker enables automatic generation of the HTML to set an initial view for an image. Involving more advanced ActionScript and numerous API calls, this example demonstrates programmatically getting values from the Zoomify Flash Component, setting values to drive the Component display, and reading and writing values to and from user interface elements. Available for download at *http://www.zoomify.com/downloads/*.

Chapter V. Slideshows, Animations, and Hotspots

Working with Slideshows

You can create slideshows of zoomable images with the Zoomifyer for *Flash* slideshow component. Additionally, you can create zoomable 3D object movies and panoramas. As with all Zoomify content, these special media types can be integrated right into your pre-existing Flash MX or MX 2004 movies.[12]

In Flash MX 2004, you'll have access to the ZoomifySlideshow component right in the Components panel under the Zoomify Components header. You also notice the ZoomifyHotspot component in the same section.

In Flash MX, you'll need to locate the Components panel and select "Zoomify Components 2" from the dropdown. You'll then be able to drag the Slideshow and Hotspot components into your movie.

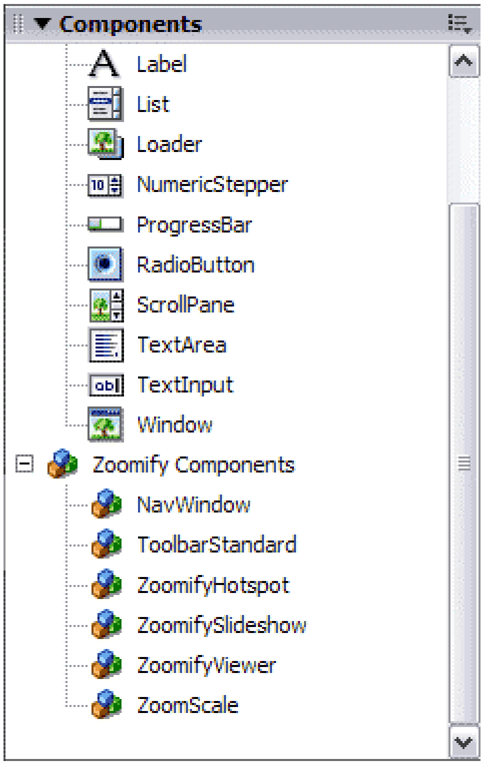


Figure 8: Flash MX 2004 Components Panel

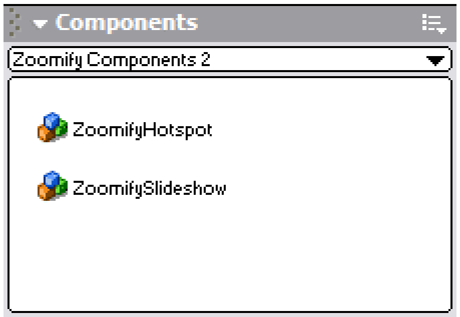
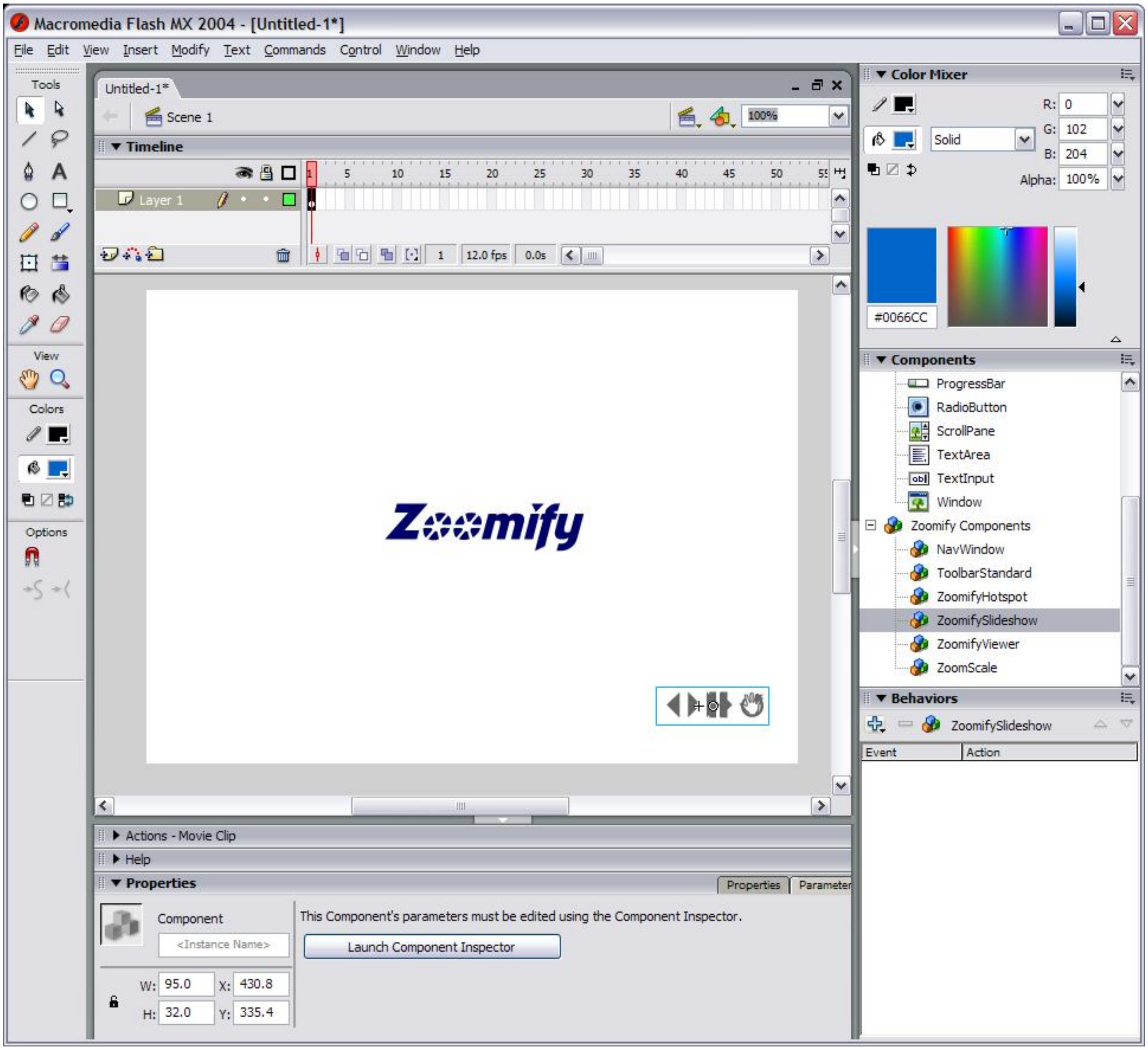


Figure 9: Flash MX Components Panel

As an initial matter, you'll need to add the ZoomifyViewer component and the

[12] Please note that many time saving techniques are available to Flash MX 2004 users. In Flash MX you can create slideshows, animations, and hotspots, but manual image conversion, manual entry of Development folder paths, and the like will be required.

ZoomifySlideshow component to your stage. First, drag the ZoomifyViewer component onto the stage and give it a unique instance name in the Flash Properties panel. Then, you may drag out the ZoomifySlideshow component and place it where you'd like it to appear within your movie. You'll then want to save your FLA immediately before continuing.[13]



Color Mixer: R: 0, G: 102, B: 204, Alpha: 100%, Solid, #0066CC

Component <Instance Name>

This Component's parameters must be edited using the Component Inspector.

W: 95.0 X: 430.8
H: 32.0 Y: 335.4

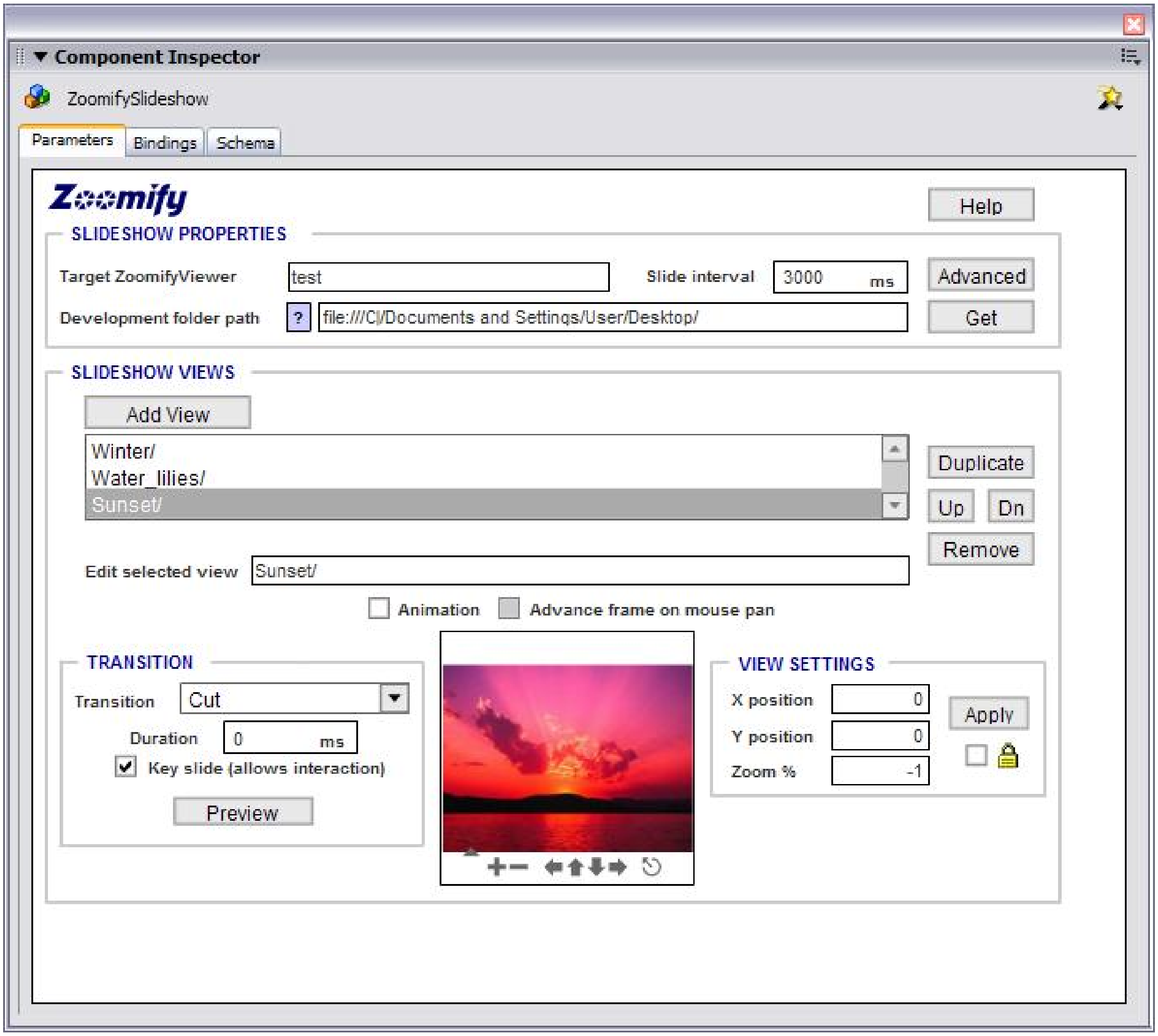
Figure 10: Flash MX 2004 stage with ZoomifyViewer and ZoomifySlideshow components.

Unlike a simple ZoomifyViewer implementation where you add an image in the ZoomifyViewer Component Inspector, you will instead select the ZoomifySlideshow component and launch its Component Inspector (located in the Properties panel). You

[13] This will enable you to utilize Zoomify's convert functionality and other features directly from Flash MX 2004, and also allow you to preview your project within both Flash MX and MX 2004.

need not add any data whatsoever to the ZoomifyViewer component itself other than an instance name.[14]

Once you've opened the Component Inspector, you'll need to associate the ZoomifySlideshow component with the ZoomifyViewer component by typing the ZoomifyViewer component's instance name into the topmost field of the ZoomifySlideshow Component Inspector (called "Target ZoomifyViewer").



SLIDESHOW PROPERTIES

Target ZoomifyViewer: test

Slide interval: 3000 ms

Development folder path: file:///C|/Documents and Settings/User/Desktop/

SLIDESHOW VIEWS

Winter/
Water_lilies/
Sunset/

Edit selected view: Sunset/

□ Animation □ Advance frame on mouse pan

TRANSITION

Transition: Cut

Duration: 0 ms

☑ Key slide (allows interaction)

VIEW SETTINGS

X position: 0

Y position: 0

Zoom %: -1

□

Figure 11: Flash MX 2004 ZoomifySlideshow Component Inspector.

Next, you can change the "Slide interval" from the default 3000 ms (3 seconds). This is the length of time that will pass before the next slide in the series loads, provided that the end user has selected the auto-advance button. Please note that the Slide interval and the Transition duration are not additive – if your Transition duration is 5 seconds, and your

[14] An instance name for the ZoomifyViewer is always required if you wish to associate any other Zoomify components or Flash components with the viewer.

Slide interval is 5000 ms or less, the slide will advance immediately upon the completion of the entrance transition.

If you'd like to preview your slideshow in the small preview view window, you'll need to enter your "Development folder path". If you've already saved your project FLA file, you can use the "Get" button to automatically load the Development folder path (Flash MX 2004 only).

The next step is to add images to your slideshow. This is done with the "Add View" button. If you are using Flash MX 2004, you can convert an image at this stage, or browse to a previously converted Zoomify Image folder. If you are using Flash MX, you'll need to manually enter the path to the image itself. Please note that in order for previews to work, you'll have to save your images within your Development folder path. Absolute URLs may also be used.

You may also use the "Duplicate" button to duplicate images already added. This can be useful if you are using an interstitial image, or if you are using the "Zoom" transition within the same image. You may also move images up and down within the slideshow list by using the "Up" and "Dn" buttons. To remove an image from the list, select it and click the "Remove" button. If you'd like to edit the pathname of a particular view, select a slide and then edit its pathname in the "Edit selected view" text field.

Next, you may add transitions to each frame of your slideshow. To do this, select an image in the slideshow list, and then select a "Transition" from the dropdown list. Transitions will be used to move to that image, and, if the end user clicks the "Previous" button on the slideshow, to transition to the previous image. You may also assign a transition time to each transition. The default, except for "Cut", is 500 ms (.5 seconds). Finally, if you want to prevent the user from zooming and panning a particular slide, uncheck the "Key slide" checkbox.

In the "Initial View" section, you can set the X, Y, and zoom levels of each slide's initial appearance. You can use the toolbar in the view area to navigate to the view you'd like to display when that slide is first viewed. The "Lock" checkbox allows you to save desired settings while still navigating the image. To move the image back to the locked settings, simply click "Apply".

To see how a particular transition will unfold, click the "Preview" button. To test the entire slideshow, use "Control > Test Move" from the Flash MX or MX 2004 menu.

Animations: Objects and Panoramas

Zoomifyer for *Flash* also enables you to create zoomable 3D objects and panoramas. It's an easy matter – just add the frames of your object or panorama via the ZoomifySlideshow Component Inspector interface, and then select the "Animation" checkbox (beneath the "Edit selected view" field).

To make adding images easier, we recommend that you name your object or panorama frames sequentially, with a number at the end. For example, you could name your object frames "Object01" through "Object14". Then, add the first frame, duplicate it 13 times, and edit each name in the "Edit selected view" field.

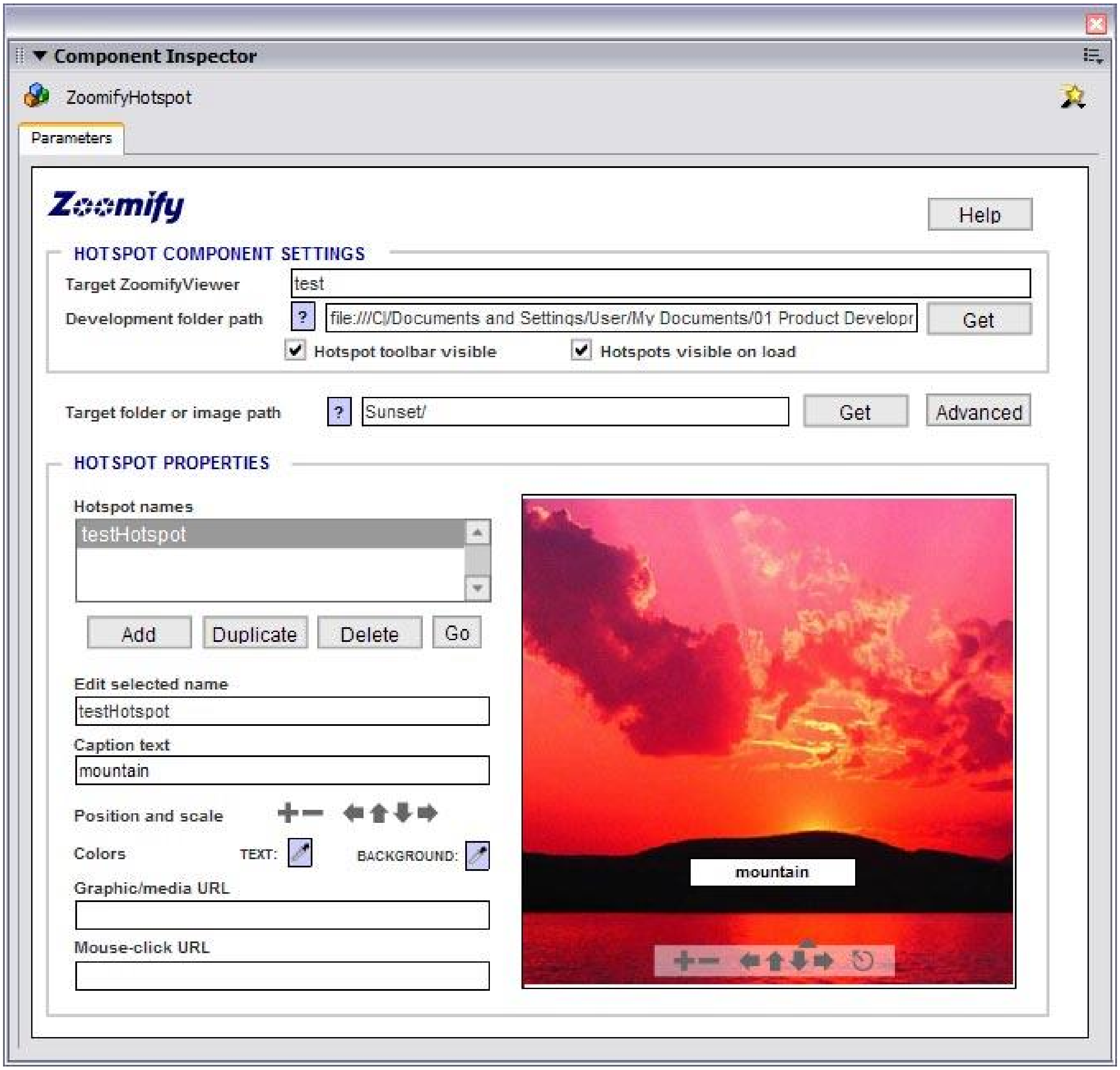
If you'd like to allow the end user to advance the animation with the mouse, ensure that the "Advance frame on mouse pan" button is checked.

Hotspots

You can also easily add hotspots to your Zoomify project. Add a ZoomifyViewer to your Flash MX or MX 2004 stage and give it an instance name. Then, drag on the "ZoomifyHotspot" component from the Components panel. Save the project so you can view previews. Then, make sure that the ZoomifyHotspot icon is selected, and then click the "Component Inspector" button in the Properties panel.

Simply fill in the field "Target ZoomifyViewer" with the instance name of your ZoomifyViewer component. In the image above, the instance name is "Test". Then, in Flash MX 2004, click the "Get" button to get the Development folder path. In Flash MX, you'll need to manually enter the development folder path.

You can also set the hotspot toolbar and the hotspots themselves visible or invisible by checking the appropriate checkboxes underneath the Development folder path text field. Note that the hotspot toolbar will show and hide the hotspots themselves. Setting the hotspot toolbar invisible is particularly useful when you're working with slideshows and animations, as you'll need to add multiple hotspot components (one per slide or frame), but likely will only want one toolbar visible.



Component Inspector

ZoomifyHotspot

Parameters

HOTSPOT COMPONENT SETTINGS

Target ZoomifyViewer: test

Development folder path: file:///C|/Documents and Settings/User/My Documents/01 Product Developr

☑ Hotspot toolbar visible ☑ Hotspots visible on load

Target folder or image path: Sunset/

HOTSPOT PROPERTIES

Hotspot names: testHotspot

Edit selected name: testHotspot

Caption text: mountain

Position and scale

Colors TEXT: BACKGROUND:

Graphic/media URL:

Mouse-click URL:

Figure 12: Flash MX 2004 ZoomifyHotspot Component Inspector.

The next text field, entitled "Target folder or image path" is used to determine which image the hotspot(s) will appear on. If you're creating a single image Zoomify project, then you can fill this in via the "Get" button in MX 2004 – the image referenced by the ZoomifyViewer Component Inspector will be automatically filled in. If you are placing a hotspot on a slideshow, refer to the slide or frame of the slideshow.[15]

[15] Note: if you are adding hotspots to a slideshow, and have duplicated image views in the slideshow, the hotspots will appear on all duplicated slides (provided you haven't edited the slide's path name). If you would like the hotspots to only appear on one particular view of a slide, you'll need to create a copy of the image folder itself and name it differently.

Once you've selected the image or frame you'd like to add a hotspot to, simply click the "Add" button, then give the hotspot a name in the "Edit selected name" field. Hotspot names are used internally and may be called anything as long as the name is unique.

If you'd like to display a hotspot caption, enter some caption text in the "Caption text" field. This is the text that will actually be displayed on the above-referenced image. If you do not wish to have a caption, you may delete the text in the field and the hotspot will not be displayed. You should then specify a "Graphic/media URL" to ensure that something is displayed.

The image itself can be zoomed and panned to offer you a better view of the image display area. The toolbar on the image display window is used for this purpose. Note that this does not alter the appearance of the image itself when it is first displayed (that is accomplished in the ZoomifyViewer or ZoomifySlideshow Component Inspector).

The "Position and scale" toolbar is used to increase or decrease the hotspot's size and move it to a desired location on the image. You may also click and drag the hotspot itself in the display view area. The "Colors" section enables you to change the appearance of the Caption text, and is only used in the event that you wish a caption to be displayed.

Next, if you wish to display a graphic or a Flash movie as a hotspot, you can specify it in the "Graphic/media URL" text field. These URLs can be either absolute or relative to the Development folder path. Note, however, that only non-progressive JPEG images will work in this field. Flash cannot display progressive JPEGs, GIFs, or other image formats. A Flash movie may be used as well.

Finally, you may associate a click-to URL with the hotspot – simply enter it in the "Mouse-click URL" field at the bottom of the ZoomifyHotspot Component Inspector.

One final note: if you are planning on using hotspots within a Flash movie that is larger than the view area of the ZoomifyViewer, you may wish to embed fonts so as to prevent hotspot labels from floating above the mask and displaying outside the ZoomifyViewer area. To do this, simply select the text in the Library, select "Character" on the Properties Panel, and select the appropriate parts of the font to embed. This does increase the size of the SWF somewhat, so this option should only be used as necessary.

Chapter VI. Advanced Features

Modifying "Zoomify Components.fla"

The Zoomify dynamic SWF supports passing parameters into the (integrated) Zoomify Flash Component through use of a series of functions in Frame 1 of the container movie. These functions receive the passed parameters and set component variables and/or call component functions. These functions can easily be used in any movie in which one or more instances of the Zoomify Flash Component are present.

Notes regarding function modifications:

As a general rule, Zoomify sets the component properties directly – and sets URL-related properties last. API calls should not be made until the image has initialized. The Init callback is used to indicate when calls to the image can be made safely. Prior to this, the properties should be used directly.

More information about passing parameters into components is available on the Macromedia Web site at:

http://www.macromedia.com/support/flash/ts/documents/flashvars.htm.

In order to ease the introduction of Flash designers and developers to the Zoomify Flash Component source code, the functions within the "Zoomify Component.fla" are clearly grouped as "Public Methods" and "Private Methods". Modifications to the Public Method functions should be undertaken with caution. Modifications to the Private Method functions are recommended only for advanced ActionScript programmers.

Zoomify API Reference

You can also access code hints and the complete Zoomify API Reference from within Flash MX or MX 2004. The Zoomify API Reference within Flash MX or MX 2004 includes detailed descriptions, examples, and additional usage notes. The following is an overview intended to familiarize you with Zoomify's API. More detailed information is available within Flash MX (from the "*Window > Reference*" menu) or MX 2004 (from the "*Help > ActionScript Dictionary*" menu).

The following are the Public Methods:

1. *MODIFYER METHODS*

    a. setView

        Sets x, y, and zoom coordinates for the current view. The values x and y will be between -1 (left edge of image) and 1 (right edge of image). The zoom value is used to specify the magnification of the image where 100 means 100% magnification. This representation is 'resolution independent.' These values can be obtained for any view by calling getX and getY, or by using the Zoomify HTML Maker Example. Special case: setting zoom to -1 fits the image exactly within the display area. Calls to setView should be followed by a call to updateView in order to cause the ZoomifyViewer component to draw the new view.

b. updateView

Causes the Zoomify Viewer to render the view. This must be called after calling setView, setX, setY, setZoom, setImagePath, setByteHandlerURL, setMapFilePath, or setSize.

c. setX

Sets the x value of the current view. The values x and y will be between -1 (left edge of image) and 1 (right edge of image). This representation is 'resolution independent.' These values can be obtained for any view by calling getX and getY, or by using the Zoomify HTML Maker Example. Calls to setX should be followed by a call to updateView in order to cause the ZoomifyViewer component to draw the new view.

d. setY

Sets the y value of the current view. The values x and y will be between -1 (left edge of image) and 1 (right edge of image). This representation is 'resolution independent.' These values can be obtained for any view by calling getX and getY, or by using the Zoomify HTML Maker Example. Calls to setY should be followed by a call to updateView in order to cause the ZoomifyViewer component to draw the new view.

e. setZoom

Sets the zoom value of the current view. The zoom value is used to specify the magnification of the image where 100 indicates 100% magnification. Special case: setting zoom to -1 fits the image exactly within the display area. Calls to setZoom should be followed by a call to updateView in order to cause the ZoomifyViewer component to draw the new view.

f. zoomIn

Used to increase the zoom value of the current view when doing so in a continuous manner. For example, when implementing a button that, if pressed and held, is intended to zoom continuously, use zoomIn.

g. zoomOut

Used to decrease the zoom value of the current view when doing so in a continuous manner. For example, when implementing a button that, if pressed and held, is intended to zoom continuously, use zoomOut.

h. panUp

Used to increase the y value of the current view when doing so in a continuous manner. For example, when implementing a button that, if pressed and held, is intended to pan upward continuously, use panUp.

i. panDown

Used to decrease the y value of the current view when doing so in a continuous manner. For example, when implementing a button that, if pressed and held, is intended to pan downward continuously, use panDown.

j. panLeft

Used to decrease the x value of the current view when doing so in a continuous manner. For example, when implementing a button that, if pressed and held, is intended to pan left continuously, use panLeft.

k. panRight

Used to increase the x value of the current view when doing so in a continuous manner. For example, when implementing a button that, if pressed and held, is intended to pan right continuously, use panRight.

l. panStop

Used after panDown, panUp, panLeft, and panRight when continuous panning of the current view is complete. Call panStop to end continuous panning. Calls to panStop should be followed by a call to updateView in order to cause the ZoomifyViewer component to draw the new view.

m. resetView

Sets the view to the initial view coordinates. A subsequent call to updateView is not necessary.

n. setInitialX

Sets the x value of the initial view, that is, the view on load and on reset. The value x will be between -1 (left edge of image) and 1 (right edge of image). This representation is 'resolution independent.' These values can be obtained for any view by calling getX and getY, or by using the Zoomify HTML Maker Example.

o. setInitialY

Sets the y value of the initial view, that is, the view on load and on reset. The value y will be between -1 (left edge of image) and 1 (right edge of image). This representation is 'resolution independent'. These values can be obtained for any view by calling getX and getY, or by using the Zoomify HTML Maker Example.

p. setInitialZoom

Sets the zoom value of the initial view, that is, the view on load and on reset. The zoom value is used to specify the magnification of the image where 100 means 100% magnification. Special case: setting zoom to -1 fits the image exactly within the display area.

q. setMaxZoom

Sets the maximum allowable zoom. The zoom value is used to specify the magnification of the image where 100 means 100% magnification. Special case: setting zoom to -1 fits the image exactly within the display area.

r. setMinZoom

Sets the minimum allowable zoom. The zoom value is used to specify the magnification of the image where 100 means 100% magnification. Special case: setting zoom to -1 fits the image exactly within the display area.

s. setSize

Compensates for changes to ZoomifyViewer width and height to avoid image distortion or skewing. Changing the size of the component on the stage is intended to change the size of the display area, not the size of the image displayed. Calling setSize with the new component width and height as inputs will ensure no image scaling occurs (false zoom), but rather, that more or less of the image fits in the display, as appropriate. Calls to setSize should be followed by a call to updateView in order to cause the ZoomifyViewer component to draw the new view.

t. zoomToView

Implements continuous zoom and pan as necessary to traverse the image from the current view x, y, and zoom coordinates to the specified view coordinates. The values x and y will be between -1 (left edge of image) and 1 (right edge of image). The zoom value is used to specify the magnification of the image where 100 means 100% magnification. This representation is 'resolution independent.'

These values can be obtained for any view by calling getX and getY, or by using the Zoomify HTML Maker Example. The duration is specified in milliseconds and is the total period of the zoom. The intervalRate is the pace at which the view should be updated and is also expressed in milliseconds. Callback is a string specifying a userdefined method to call on arrival at the specified view and callback data is a value to pass to that method.

u. stopZoomToView

Ends any zoomToView already in progress.

v. setEnabled

Enables or disables mouse and keyboard support for the target ZoomifyViewer instance.

w. setFocus

Sets focus on a specified ZoomifyViewer instance. When called with a true value, enables the target instance to receive key events. If a focus callback has been registered, setting focus to true will trigger the callback. Note that focus is automatically set to true by mouse interaction with the image.

2. *STATUS METHODS*

a. getX

Returns the x coordinate of the current view. This value can be applied using setX or setView, followed by updateView.

b. getY

Returns the y coordinate of the current view. This value can be applied using setY or setView, followed by updateView.

c. getZoom

Returns the zoom value (percent magnification) of the current view. This value can be applied using setView or setZoom.

d. getInitialX

Gets the x value of the initial view, that is, the view on load and on reset. The value x will be between -1 (left edge of image) and 1 (right edge of image). This representation is 'resolution independent.'

e. getInitialY

Gets the y value of the initial view, that is, the view on load and on reset. The value y will be between -1 (left edge of image) and 1 (right edge of image). This representation is 'resolution independent'.

f. getInitialZoom

Gets the zoom value of the initial view, that is, the view on load and on reset. The zoom value is used to specify the magnification of the image where 100 means 100% magnification. Special case: a zoom setting of -1 fits the image exactly within the display area.

g. getMaxZoom

Gets the maximum zoom permitted. The zoom value is used to specify the magnification of the image where 100 means 100% magnification. Special case: a maximum zoom setting of -1 limits zoom-in to a level such that the image fits exactly within the display area.

h. getMinZoom

Gets the minimum zoom permitted. The zoom value is used to specify the magnification of the image where 100 means 100% magnification. Special case: a minimum zoom setting of -1 limits zoom-out to a level such that the image fits exactly within the display area.

i. getViewHeight

Returns the height of the ZoomifyViewer instance in pixels. Use getImageHeight in place of reading the component property _height as this property is not designed to allow for zoom and pan effects and will include vertical distance encompassing image content that is zoomed out of view.

j. getViewWidth

Returns the width of the ZoomifyViewer instance in pixels. Use getViewWidth in place of reading the component property _width as this property is not designed to allow for zoom and pan effects and will include horizontal distance encompassing image content that is zoomed out of view.

k. getImageHeight

Returns the height of the image at 100% zoom (the height of the original source image) in pixels.

l. getImageWidth

Returns the width of the image at 100% zoom (the height of the original source image) in pixels.

m. getEnabled

Returns whether mouse and keyboard support for the target ZoomifyViewer instance are currently enabled or disabled.

n. getFocus

Gets whether focus is currently true for a specified ZoomifyViewer instance. When a ZoomifyViewer instance has focus, key events are received. Note that focus is automatically set to true by mouse interaction with the image.

o. calculateScreenZoom

Returns the zoom level necessary to exactly fit the image within the display area.

p. checkTileQueue

Returns whether image tile requests are currently in the request queue. Tiles remain in the request queue until retrieved from the cache or received from the server.

3. PATH AND URL METHODS

a. setImagePath

Sets the path for the target Zoomify Image folder or file for a ZoomifyViewer instance. The path must be relative to the root of the server. Calls to setImagePath should be followed by a call to updateView in order to cause the ZoomifyViewer component to draw the new view.

b. getImagePath

Gets the path for the target Zoomify Image folder or file for a ZoomifyViewer instance.

c. setByteHandlerURL

Sets the path for the byte handler for a ZoomifyViewer instance if a byte handler has been specified. The byte handler is a Java servlet or CGI file that fulfills requests from the ZoomifyViewer component for header information, offsets, or tiles. A byte handler is used when publishing with JPEG2000 or PFF image files and is not required when publishing with Zoomify images folders. Calls to setByteHandlerURL should be followed by a call to updateView in order to cause the ZoomifyViewer component to draw the new view.

d. getByteHandlerURL

Gets the path for the byte handler for a ZoomifyViewer instance if a byte handler has been specified.

e. setMapFilePath

Sets the path for the map file for a ZoomifyViewer instance if a map file has been specified. A map file is used to specify virtual directory names for image locations. The path that is passed into this function must be relative to the root of the server. Calls to setMapFilePath should be followed by a call to updateView in order to cause the ZoomifyViewer component to draw the new view.

f. getMapFilePath

Gets the path for the map file for a ZoomifyViewer instance if a map file has been specified.

4. CALLBACK METHODS

a. registerCallback

Registers for a callback of a certain type. The name of the callback function to be called is passed in as a string. Callback values include:

1. "Init" - called when the image is done initializing. An image is initialized once it has downloaded it's header information.

2. "DownloadComplete" - called when the download queue becomes empty. Useful after a call to updateView to determine when all tiles have been received.

3. "Mouse" - called on mouse down and mouse up in the viewer. The callback is called with 2 parameters. The first parameter is the instance pointer (this) and the second parameter: "start" on mouse down or "stop" on mouse up.

4. "Key" - called on key down and key up. The callback accepts 3 parameters. The first parameter is the instance pointer (this), the second is either "down" or "up" and the third is the keyCode of the pressed key. This callback is only triggered for keystrokes that the ZoomifyViewer responds to.

5. "Focus" - called whenever focus of this instance is set to true. This can happen through either a setFocus call or use interaction with the ZoomifyViewer instance.

6. "View" - called whenever the view changes. The view is reported as x,y, and zoom. See setView for a description of these parameters.

7. "Rect" - called whenever the view changes. In this case, the view is reported back as a bounding box in platform independent coordinates. This is useful for drawing a navigation window that shows where the user is currently looking.

8. "MaxZoomReached" - called when the max zoom is arrived at while zooming with key or zoomIn function.

9. "MinZoomReached" - called when the min zoom is arrived at while zooming with key or zoomIn function.

10. "Status" - called when the ZoomifyViewer engine report status.

b. unregisterCallback

Clears a specified, previously registered callback.

5. LABEL METHODS

a. addLabel

Adds a label to the display list and the current view. Accepts an XML node that describes a label. Labels have an x, y, zoom, xscale, yscale, library movie clip, and url to launch when clicked.

b. setLabelProperty

Given a label ID and property name, changes the value of that property. Valid properties include x, y, xscale, yscale, and url.

c. setLabelVisibility

Sets the visibility of all labels.

d. getLabelVisibility

Gets the visibility of all labels.

e. removeLabel

Removes a label from the display list and the current view. Note that this is not equivalent to setting label visibility to false. Reversing label removal requires explicitly re-adding the label.

f. removeAllLabels

Removes all labels. Note that this is not equivalent to setting label visibility to false. Reversing label removal requires explicitly re-adding the label.

Thumbnail Support

The Zoomifyer for *Flash* solution is completely compatible with common web site designs involving a simple thumbnail that links to a pop-up providing a more detailed view.

To implement this site design using any Zoomify Image folder simply refer to the first JPEG tile within the image folder. This file will be named "0-0-0.jpg" and will be located in the folder "TileGroup0". It will be between 128 and 256 pixels in height and width - a common scale for web site thumbnails. A standard JPEG reference to this file can easily be embedded within any HTML page. For example:

```
 <img src = "../imageName/TileGroup0/0-0-0.jpg" border="0">
```

Chapter VII. Troubleshooting

Troubleshooting installation

If you find that installation returns an error message, is interrupted, or that certain files are not installed, you might check the following:

1. Did you first install Macromedia Flash MX or MX 2004 and run it at least once before attempting installation of Zoomifyer for *Flash*? It is essential that this is done first as this will ensure that they proper file structure has been created.

2. Was Flash MX or MX 2004 running at the time you attempted installation? This might prevent certain files from being written to their proper directories. Please quit Flash MX or MX 2004 and then reattempt installation of Zoomifyer for *Flash*.

In the event that you cannot install the Zoomifyer for *Flash* files via the provided installer application, you can install the files manually. Please see the appendix entitled "Manual Installation of Zoomifyer for *Flash*" for instructions.

Troubleshooting image encoding

1. If you receive an error message or the Zoomifyer unexpectedly quits, you may wish to check the following:

   a. Does the directory you're attempting to write the Zoomify image to (the target directory) have enough disk space? Generally the Zoomifyer requires a substantial amount of swap space when encoding an image. This swap space is freed up after successful conversions.
   b. Is the target directory writeable (i.e., not a CD or a directory / disk with limited write permissions). If so, select a target directory that is writeable before proceeding.
   c. Are you trying to convert a non-native format to the Zoomify format and do not have QuickTime 5 or higher installed? The Zoomifyer supports TIFF, BMP, and JPEG natively – otherwise it relies on the QuickTime image libraries. If these are not present, conversion will not occur.

2. If conversion is slower than you expected:

   a. Are you running the Zoomifyer on an older OS such as OS 9 or Windows 95/98/ME? We strongly recommend that you run the Zoomifyer on OS X or Windows 2000/XP.
   b. Are you trying to convert an image that is not native to the Zoomifyer? The Zoomifyer supports TIFF, BMP, and JPEG images natively. Other formats rely on QuickTime's image libraries. Although the Zoomifyer is able to employ the

QuickTime libraries, making use of them is generally much slower than native formats.[16]

c. Is your source image on a network drive? We recommend that you copy all images you'd like to encode to a local drive.
d. Is your target directory a network drive? As the Zoomifyer encoding process is disk-intensive, we recommend that you convert to a local drive.
e. Do you have enough memory?

3. If you still experience problems encoding, please send an email to support@zoomify.com with details about your environment, including OS, amount of free disk space, available RAM, Zoomifyer product and version number, the source image format and size, and any other information you believe to be relevant. If we determine that the issue is related to the image itself and not the environment, we may request the image for testing purposes.[17]

[16] For optimal results working with source images in an uncompressed (BMP, TIFF) or 'losslessly' compressed (PNG) source images is recommended rather than working with images in a 'lossy' format such as the popular JPEG format. The Zoomifyer itself will apply JPEG compression to the tiles created during conversion. When working from lossy compressed source images this would effectively represent a second lossy compression and would introduce the possibility of visible artifacts if caution is not exercised.

[17] Zoomify respects intellectual property rights and any image provided to Zoomify will not be used for any other purpose than testing unless otherwise indicated by the copyright holder.

Chapter VIII. Appendices

Using Zoomifyer for Flash *with Flash MX*

Zoomifyer for *Flash*, though designed to take advantage of the feature-set of Macromedia Flash MX 2004, will also work well with Macromedia Flash MX. Please note that installation must be done manually, and not all of the features described above will be present.

For example, in Flash MX you cannot convert images into Zoomify Images from within Flash MX itself. Rather, images must be converted via the alternative droplet method. Also, the Zoomify Behavior functions are not available, nor are certain dialogs such as the browse dialog.

Complementary Zoomify products – available at www.zoomify.com

1. **Zoomifyer *Enterprise***

   Zoomifyer *Enterprise* delivers mission-critical functionality essential to microscopy and other high-resolution imaging applications involving multi-gigabyte images as well as integrated implementations requiring server-side automation.

   Zoomifyer *Enterprise* is designed for:

   a. higher-education image gallery solutions
   b. telepathology and medical research
   c. industrial quality assurance
   d. semiconductor inspection
   e. materials science and metallurgy
   f. pharmaceutical chemistry & biotechnology

   Features include:

   a. server-side image conversion support for batch processing and automation
   b. online annotation including non-destructive icon and text capabilities
   c. conversion support for multiple source images
   d. optional lossless encoding
   e. optional image decoding to editable BMP format
   f. decoding support for images stored in the LizardTech MrSid format or the MGI FlashPix format.
   g. 1 year maintenance contract (phone and email support, issue tracking, and updates)

   For more information, please visit: *http://www.zoomify.com/enterprise/.*

Manual Installation of Zoomifyer for Flash

If you experience any difficulties installing Zoomifyer for *Flash* with the provided installer application, you may manually install the files by following these instructions.

First, ensure that you've already installed Macromedia Flash MX 2004 or Flash MX and run it at least once. This will ensure that the necessary folder structure is created.

There are two distinct directories where content will be copied. The first is user-dependent and therefore the Zoomifyer for *Flash* files must be copied to the current user's profile. The second group of files goes in the "Program Files" (Windows) or "Applications" (Mac OS X) directory. Also, the following presumes an English installation. Please see below for the appropriate folder for other languages. Also, the locations will vary depending on whether you have Flash MX or Flash MX 2004.

Zoomifyer for *Flash* comes with 2 MXI files located within the "Developer Tools/Installer Support Files" folder. The first, called "Zoomify.mxi" is designed for Macromedia Flash MX 2004. The second, called "ZoomifyMX.mxi", is designed for Macromedia Flash MX. These can be opened in a text editor and will provide locations for all files to be installed.

Manual Installation for Flash MX

Copy the following files into the following OS-dependent folders:[18]

- Windows 2000 and XP

```
C:\Documents and Settings\[USER]\Application Data\Macromedia\Flash MX\
Configuration\
```

- Windows 98

```
C:\Windows\Local Settings\Application Data\Macromedia\Flash MX\Configuration\
```

- MAC OS X

```
Hard Drive:Users:Library:Application Support:Macromedia:Flash
MX:Configuration:
```

The files to copy:

Feature: Zoomify API Reference (like the ActionScript Dictionary)
Files: Tools/Zoomify Reference.xml (Custom Actions folder)

[18] Note that the c:\ drive is a placeholder and may vary depending on your system setup. Also, in Windows 2000 and XP, the [user] folder will be the same as that of Flash MX or MX 2004 installation (i.e., the currently logged-on user).

Destination folder: \ActionsPanel\CustomActions\[19]

Feature: Zoomify Components (ZoomifyViewer, ToolbarStandard, NavWindow, ZoomScale, ZoomifySlideshow, ZoomifyHotspot)
File: Zoomify Components.fla (Zoomify Components folder)
File: Zoomify Components 2.fla (Zoomify Components folder)
Destination folder: \Components\

Feature: Zoomify Examples
Files: Zoomify Examples.swf and Examples folder contents (1 file and 8 subfolders with many files, see file list above for detail)[20]
Destination folder: \WindowSWF\Zoomify Examples\[21]

Manual Installation for Flash MX 2004

Copy the first set of files into the following OS-dependent folders:

- Windows 2000 and XP

```
C:\Documents and Settings\[User]\Local Settings\Application
Data\Macromedia\Flash MX 2004\en\Configuration\
```

- Windows 98

```
C:\Windows\Local Settings\Application Data\Macromedia\Flash MX
2004\en\Configuration\
```

- MAC OS X

```
Hard Drive:Users:Library:Application Support:Macromedia:Flash MX
2004:en:Configuration:
```

First set of files for Flash MX 2004 include:

Feature: Zoomify Components (ZoomifyViewer, ToolbarStandard, NavWindow, ZoomScale, ZoomifySlideshow, ZoomifyHotspot)
File: Zoomify Components.fla (Zoomify Components folder)
File: Zoomify Components 2.fla (Zoomify Components folder)
Destination folder: \Components\

Feature: Zoomify Behavior (integrated zoom functionality)
File: Zoomify_Viewer.xml (Behavior folder)
Destination folder: \Behaviors\

[19] As an example, you would copy the file "Zoomify Reference.xml" from the Zoomifyer for *Flash* folder "Developer Tools\Custom Actions" into "C:\Documents and Settings\[USER]\Application Data\Macromedia\Flash MX\Configuration\Actions Panel\CustomActions\".

[20] The "Zoomify Examples.swf" file is located in the \Developer Tools\Installer Support Files\ directory. The "Examples" folder is located in the \Documentation\ directory.

[21] You'll have to create the folder called "Zoomify Examples".

Feature: Zoomify Command (integrated image conversion)
File: Zoomify Converter.jsfl (Commands folder)
Destination folder: \Commands\

Feature: Zoomify code hints
File: Zoomify Custom Actions.xml (Custom Actions folder)
Destination folder: \ActionsPanel\CustomActions\

Feature: Zoomify API Reference (like the ActionScript Dictionary)
Files: ZoomifyAPIReference folder (66 files included, see file list above)
Destination folder: \HelpPanel\Help\

Feature: Zoomify integrated features including Zoomify registration and support html page, converter application, and file access DLL file
Files: Register-Support-Upgrade.htm, Zoomify Converter.exe, ZoomifyFileAPI.dll[22]
Destination folder: \External Libraries\

Copy the second set of files into the following OS-dependent folders:

Windows 2000 and XP and 98

```
C:\Program Files\Macromedia\Flash MX 2004\en\Configuration\
```

Mac OS X

```
Hard Drive:Applications:Macromedia:Flash MX 2004:Configuration:
```

Second set of files:

Feature: Zoomify Examples
Files: Zoomify Examples.swf and Examples folder contents (1 file and 8 subfolders with many files, see file list above for detail)[23]
Destination folder: \HelpSWF\Zoomify Examples\[24]

The folder /en/ is representative of an English installation.

According to the latest information at the time of release, the following folder names correspond to the following languages:

Language Folder Name

[22] The file "Register-Support-Upgrade.htm" file is located in the \Documentation\ folder. The file "Zoomify Converter.exe" is located at the root of the Zoomifyer for *Flash* download folder. The "ZoomifyFileAPI.dll" file is located in the \Developer Tools\Installer Support Files\ folder. Note that .exe files are called .app files on Mac OS X.
[23] The "Zoomify Examples.swf" file is located in the \Developer Tools\Installer Support Files\ directory. The "Examples" folder is located in the \Documentation\ directory.
[24] You'll have to create the folder called "Zoomify Examples".

English	en
French	fr
German	de
Italian	it
Japanese	ja
Portuguese	pt
Spanish	es
Swedish	sv
Simplified Chinese	zh_cn
Traditional Chinese	zh_tw
Korean	ko

Note that the Zoomify for *Flash* installer is designed to appropriately handle these language settings.

Uninstalling Zoomifyer for Flash

Uninstalling Zoomifyer for *Flash* is easy – simply delete the Zoomify-specific files. As Zoomifyer for *Flash* is integrated into Flash MX or MX 2004, it does not impact the OS.

Glossary

ActionScript: Macromedia's Flash-based scripting language, similar to JavaScript.

Compression: Sophisticated means to reduce file size while retaining file information by substituting representations or approximations of the file data for the actual data.

FLA: Macromedia's editable Flash format.

Flash MX 2004: The latest version of Macromedia's tool for creating rich Internet content and applications.

JRun 4: The latest version of Macromedia's Java application server.

Lossy: Compression that permanently removes image data to reduce file size. Lossless compression, by comparison, keeps all original image data intact.

Metadata: Additional information embedded in an image format (such as a TIFF). May include copyright, author, data, and equipment information.

Multi-resolution: Data file containing many levels of quality.

PFF: Zoomify's streaming image format.

Pixel: The smallest unit of screen display. A point of color.

Pre-fetching: Intelligent retrieval of data from a Web server in anticipation immediately subsequent use.

Pyramidal image: A data file format containing multiple levels of resolution.

Resolution: The amount of image data available for display. One measure of image quality. Screen equivalent of printer 'dpi' (dots per inch).

Servlet: A small Java-based server-side application. Requires a servlet container.

Servlet container: A Java-based application that calls upon servlets to perform specific server-side tasks.

SWF: Macromedia's published Flash format.

Tile: Piece of an image at a given resolution.

Wavelet: Highly effective image compression approach enabling maximum reduction in file size while preserving image quality.

System requirements

Hardware

Apple Macintosh: PowerMac
Intel: Pentium II or better

Memory

64 megabytes of RAM required, 128 or higher recommended

Operating systems

Apple Macintosh OS: Version 9.1 or higher (viewing), OS X v10.2.6 (authoring)
Microsoft Windows: 98, ME, NT 4, 2000, XP (viewing), 98, 2000, XP (authoring)

Software applications

Macromedia Flash Player 6 or higher (viewing), Flash MX or higher (authoring)
Apple QuickTime: Version 5.0 or higher recommended for extended image libraries

Additional resources

Companies & Organizations

Zoomify, Inc.	www.zoomify.com
Macromedia, Inc.	www.macromedia.com

Software

Zoomifyer *EZ*	www.zoomify.com/ez/
Zoomifyer *Extension*	www.zoomify.com/extension/
Zoomifyer for *Flash*	www.zoomify.com/flash/
Zoomifyer *Enterprise*	www.zoomify.com/enterprise/
Macromedia Flash MX 2004	www.macromedia.com/flash/
Macromedia Flash Communication Server	www.macromedia.com/flashcom/
Aladdin Systems' StuffIt Expander	www.stuffit.com
WinZip Computing Inc.'s WinZip	www.winzip.com