Using 3D in Viewpoint Technology

Version 1.0
January 2002
Using 3D in Viewpoint Technology

Copyright, Notice of Disclaimer and Limitation of Liability

© 2003 Viewpoint Corporation. All Rights Reserved.

Using 3D in Viewpoint Technology

Viewpoint, the Viewpoint logo, Viewpoint Experience Technology (VET), Viewpoint Media Compressor, Viewpoint Media Publisher, Viewpoint FinalCheck, Viewpoint Scene Builder, and Viewpoint Media Player (VMP) are registered trademarks or trademarks of Viewpoint Corporation in the United States and in other countries.

Companies, names, and data used in examples herein are fictitious unless otherwise noted. Information in this document is subject to change without notice.

Macromedia and Flash are trademarks or registered trademarks of Macromedia, Inc. in the United States and/or other countries. All other product and company names mentioned herein are the trademarks of their respective owners.

All other product and company names mentioned herein are the trademarks of their respective owners.

Disclaimer

Except as expressly provided otherwise in an agreement between you and Viewpoint, all information, software, and documentation is provided “as is,” without warranty of any kind. Viewpoint makes no warranties, express or implied, including without limitation the implied warranties of merchantability and fitness for a particular purpose regarding such information, software and documentation. Viewpoint does not warrant, guaranty, or make any representations regarding the use or the results of the software in terms of its correctness, accuracy, reliability, timeliness, suitability or otherwise. The entire risk as to the results of performance of the software is assumed by you.

In no event will Viewpoint be liable for any special, indirect, consequential, punitive, or exemplary damages or the loss of anticipated profits arising from the performance of the software or resulting from the loss of use, data or profits, whether in an action for breach of contract or warranty or tort (including negligence) arising out of or in connection with the information, technology, software and documentation.

The Web site and publications may contain technical inaccuracies or typographical errors. Viewpoint assumes no responsibility for and disclaims all liability for any such inaccuracy, error, or omission in the Web site and documentation and in any other referenced or linked documentation. Viewpoint may make changes to the information, software, Web site, documentation, prices, technical specifications, and product offerings in its sole discretion at any time and without notice.

Author: Carolyn Gronlund and Doralee Moynihan

Contributors: Andrew Cook

Viewpoint Corporation
498 Seventh Avenue
Suite 1810
New York, NY 10018

© 2003 Viewpoint Corporation. All rights reserved.
Contents

Chapter 1: Introduction ............................................................ 6
   About This Guide ................................................................. 6
   About Viewpoint Technology ................................................... 7
   About Viewpoint Technology Web Applications ......................... 7
   Minimum System Requirements .............................................. 8

Chapter 2: Preparing Your 3D Model to Use with Viewpoint Technology .................................................. 10
   3D Authoring Applications and Related Viewpoint Technology Utilities ....... 10
   Before You Export or Save to Viewpoint Technology ...................... 11
   Exporting or Saving to Viewpoint Technology Media Files ................ 13
   Placing a Viewpoint Technology Web Application Into a Web Page ........ 14
   Testing Your 3D Model's Animation ......................................... 15
   Adjustments You Can Make After Exporting to Viewpoint Technology .... 15

Chapter 3: Adding Animations to Your Viewpoint Technology Web Application .............................................. 18
   About Animations in a Viewpoint Technology Web Application ............ 18
   Creating Simple Animations ................................................... 19
   Creating Texture Animations .................................................. 24

Chapter 4: Adding Hot Spots to a Viewpoint Technology Web Application ....................................................... 28
   What is a Hot Spot? ................................................................... 28
   Creating Hot Spots With Rollover Text ......................................... 28
   Creating a Hot Spot to Display a 2D Graphic .................................. 30
   Using a Hot Spot to Display a Flash Movie .................................... 31
   Setting the Cursor to Indicate Clickable Hot Spots ............................ 34
   Creating a Hot Spot to Display Descriptive Text ................................. 35

Chapter 5: Triggering Animations ............................................ 38
   Setting a Hot Spot to Trigger an Animation ..................................... 38
   Triggering an Animation From a Scene Object .................................. 39
   Triggering Animations From HTML Using JavaScript ....................... 40

Chapter 6: Adding Flash to a Viewpoint Technology Web Application ............................................................ 42
   Flash Movies in Viewpoint Technology Web Applications .................. 42
   Adding a Flash Movie to Your Viewpoint Technology Web Application .... 43
   Using a Flash Movie in the Foreground or Background ...................... 43
   Using a Flash Image or Movie as a Texture ..................................... 44
   Preloading a Flash Movie ......................................................... 48

Chapter 7: Testing, Troubleshooting, and Deploying .......... 52
   About Testing: A Four-Phase Approach ........................................ 52
   Uploading Your Content to a Test Web Page ................................... 52
   Testing Animations and Interactivity ............................................ 53
   Testing Viewpoint Technology Content Across Browsers and Operating Systems 54
   Troubleshooting ........................................................................ 56
   Deploying Your Viewpoint Technology Web Application ................... 57

© 2003 Viewpoint Corporation. All rights reserved.
Appendix A: Help, Resources, and Feedback ........................................ 58
  Viewpoint Developer Central: A Complete Resource ................................ 58
  Download Viewpoint Applications, Guides, and Examples ..................... 58
Glossary .................................................................................................. 61
Chapter 1: Introduction

About This Guide

This guide describes how to create a basic 3D Viewpoint Technology-enabled web application with simple interactivity and Macromedia® Flash™ effects. You'll learn how to take content that has been created in a 3D modeling application, enhance it with Viewpoint Technology, and publish it to a web page.

This guide includes the following chapters:

- **Chapter 1: “Introduction”** — Provides an overview of Viewpoint Technology, rich media, Viewpoint Technology web applications, and the authoring tools you can use to create a Viewpoint Technology web application.
- **Chapter 2: “Preparing Your 3D Model to Use with Viewpoint Technology”** — Offers guidelines for how to export or save 3D models using your preferred modeling application for use with Viewpoint Technology.
- **Chapter 3: “Adding Animations to Your Viewpoint Technology Web Application”** — Describes how to create animations with Viewpoint Technology tools.
- **Chapter 4: “Adding Hot Spots to a Viewpoint Technology Web Application”** — Gives steps for creating hot spots for your Viewpoint Technology web application.
- **Chapter 5: “Triggering Animations”** — Offers ways to trigger animations in a Viewpoint Technology web application.
- **Chapter 6: “Adding Flash to a Viewpoint Technology Web Application”** — Suggests ways to use Flash image or movie as a foreground, background, or texture in a Viewpoint Technology web application. This chapter also describes how to create a Flash preloader for a Viewpoint Technology scene.
- **Chapter 7: “Testing, Troubleshooting, and Deploying”** — Describes how to verify that a Viewpoint Technology web application works as you want it to.
- **Appendix A: “Help, Resources, and Feedback”** — Lists other Viewpoint resources available to you.
- **“Glossary”** — Defines terms used in this guide.

Using Code Samples in This Guide

This guide includes several XML code examples. To quickly implement functionality described in this user guide, you can copy code samples from this guide and paste them into your .mtx file.
To copy a code sample from the .pdf version of this guide

1. In Adobe® Acrobat® Reader, click the Text Select Tool button.

2. Click and drag to highlight the code sample you want.

3. Right-click the highlighted text, and click Copy.

About Viewpoint Technology

Viewpoint Technology is a unique technology developed by Viewpoint Technology that streams rich media over the Internet via Viewpoint Media Player™, a web browser plug-in.

Viewpoint Technology allows you to combine all types of rich media including 2D photos and drawings, 3D models, animation, ZoomView images, Flash movies, video, text, and audio to create rich media content. Using a sophisticated set of compression techniques, Viewpoint tools create a visually crisp, smooth-streaming multimedia experience for the user.

Note: To publish Viewpoint Technology content on a website, the site publisher must complete a Broadcast License Agreement and acquire the associated Broadcast License Key (a "key"). Broadcast licenses are FREE for non-commercial use. For more information, please visit the Broadcast Key section of Viewpoint Developer Central (http://developer.viewpoint.com/).

About Viewpoint Technology Web Applications

Until recently, Viewpoint technology has been described in terms of "3D scene", "rich media content", and "Viewpoint Technology-enabled web pages". Developing for the web with Viewpoint Technology, however, offers much more than those terms imply. With Viewpoint Technology, you are creating interactive Viewpoint Technology web applications that can include XML, JavaScript, and rich media such as 3D models, 2D graphics, animations, sound, and Flash files, as well as the Viewpoint proprietary ZoomView and HyperView technologies.

A Viewpoint Technology web application can be confined to the Viewpoint Technology layer (embedded in a web page) or can include the entire web page with JavaScript interactions between the HTML doc and the Viewpoint Technology scene. Web developers using Viewpoint Technology can employ a variety of content-delivery options, including interactivity and navigation, as well as a full range of rich media, including 3D models, 2D graphics, animation, and sound.
Viewpoint Technology web applications are created using standard authoring tools for 3D modeling, 2D graphics, and so on, as well as Viewpoint tools such as Viewpoint Scene Builder™ — a tool used to assemble and edit the content of a Viewpoint scene.

Using Scene Builder, a 3D modeling application's export option, and/or any XML editor, you can generate the files required to create a Viewpoint Technology web application. These are the key files in a Viewpoint Technology web application:

- **.mts file** — Contains a compressed collection of rich media components. These components are orchestrated by the .mtx file to create a scene.

- **.mtx file** — An XML-based file that contains the hierarchical relationships between elements in the scene and is the script for staging them. This file instructs Viewpoint Media Player what to do with the rich media components it finds in the related .mts files for the scene. These instructions include information about position, rotation, scale, opacity, and so on. This file can also reference external files (such as Flash .swf files), separate .mts files, and even other .mtx files.

  **Note:** Broadcast Key files also use the .mtx filename extension and are typically named BroadcastKey.mtx or bkey.mtx. If you have an old Broadcast Key file with a .txt filename extension, you should rename the file to have an .mtx extension instead and change any references to the Broadcast Key file to reflect the name change.

- **.mtz file** — The binary compressed version of an .mtx file.

- **.mzv file** — A file format for compressed image tiles (parts of a large image) used by the Viewpoint ZoomView technology, a component of Viewpoint Media Player.

  **Note:** Even though a rich media component is in an .mts file, it is not necessary to use it in the resulting scene. The .mts file is your creative arsenal. The .mtx file orchestrates the elements within a scene.

To place a rich media component in a scene, Scene Builder publishes an .mtx file containing XML tags to acquire atoms from its .mts file. Viewpoint Media Player reads the designated .mts file and extracts scene elements.

You can also store rich media components such as Flash .swf files, sound files, and JPEG textures outside a scene. In this case, the .mtx file references the .swf files to apply them to the scene.


**Minimum System Requirements**

**Scene Builder**

- Pentium® 166 MHz processor
- 128 MB RAM

© 2003 Viewpoint Corporation. All rights reserved.
• 30 MB available hard-disk space
• TrueColor display (24-bit recommended)
• 1024 x 768 monitor resolution

**Viewpoint Media Player**

<table>
<thead>
<tr>
<th>Windows</th>
<th>Macintosh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Netscape Navigator® 4.07 or later (not 6.0)</td>
<td>Netscape Navigator® 4.7 or later (not 6.0)</td>
</tr>
<tr>
<td>Microsoft® Internet Explorer 4.x or later</td>
<td>Microsoft Internet Explorer® 5.x or later</td>
</tr>
<tr>
<td>AOL® version 4, 5, 6, or 7</td>
<td>AOL® version 4 or 5 (supported Internet Explorer and Netscape Navigator browsers only)</td>
</tr>
<tr>
<td>CompuServe 7</td>
<td>256 color display (millions of colors recommended)</td>
</tr>
<tr>
<td>256 color display (24-bit recommended)</td>
<td>28.8 Kbps modem</td>
</tr>
<tr>
<td>28.8 Kbps modem</td>
<td>5 MB available hard-disk space</td>
</tr>
<tr>
<td>5 MB available hard-disk space</td>
<td>128 MB RAM</td>
</tr>
<tr>
<td>32 MB RAM (64 MB recommended)</td>
<td>PowerPC® 604 processor</td>
</tr>
<tr>
<td>Pentium® 166 MHz processor (Pentium II recommended)</td>
<td><strong>Viewpoint Media Publisher</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Windows</th>
<th>Macintosh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pentium® 166 MHz processor</td>
<td>128 MB RAM</td>
</tr>
<tr>
<td>128 MB RAM</td>
<td>30 MB available hard-disk space</td>
</tr>
<tr>
<td>30 MB available hard-disk space</td>
<td>TrueColor display (24-bit recommended)</td>
</tr>
<tr>
<td>TrueColor display (24-bit recommended)</td>
<td>1024 x 768 monitor resolution</td>
</tr>
</tbody>
</table>
Chapter 2: Preparing Your 3D Model to Use with Viewpoint Technology

3D Authoring Applications and Related Viewpoint Technology Utilities

Viewpoint Technology lets you harness the full power of the web by supporting the rich media content and interactivity you already use along with the Viewpoint 3D, HyperView, and ZoomView technologies. You can work with tools you already have, including the 3D authoring tools of your choice.

In addition, you can use the free-of-charge utilities available from Viewpoint Developer Central (http://developer.viewpoint.com) to assist you in Viewpoint Technology web authoring.

3D Modeling Applications That Support Viewpoint Technology

Viewpoint Technology files can be either saved or exported from most 3D authoring packages. See the table below for details regarding Viewpoint Technology and your 3D authoring package:

<table>
<thead>
<tr>
<th>3D Modeling Application</th>
<th>Company Name</th>
<th>Native Support¹</th>
<th>Export via Plug-in²</th>
</tr>
</thead>
<tbody>
<tr>
<td>3ds max™ (formerly 3D Studio MAX)</td>
<td>Discreet</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>3D Exploration™</td>
<td>Right Hemisphere Ltd.</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Atmosphere™</td>
<td>Adobe® Systems Inc.</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Carrara Studio™</td>
<td>Eovia</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>CINEMA 4D (Bola plug-in)</td>
<td>MAXON Computer, Inc.</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>FormZ</td>
<td>auto.des.sys, Inc.</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Geomagic Studio</td>
<td>Raindrop Geomagic, Inc.</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ImageModeler® 2.1</td>
<td>REALVIZ®</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Interactive Product Animator</td>
<td>Immersive Design</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>LightWave 3D®</td>
<td>NewTek</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Maya®</td>
<td>Alias</td>
<td>Wavfront™</td>
<td>X</td>
</tr>
<tr>
<td>Merlin 3D™</td>
<td>Digital Immersion Software Corp.</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>PolyTrans</td>
<td>Okino Computer Graphics, Inc.</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
Preparing Your 3D Model to Use with Viewpoint Technology

1 For information about getting the Viewpoint Technology Media Exporter plug-in for your 3D modeling application, see to the Authoring Tools section of Viewpoint Developer Central. Click the name of your 3D authoring tool to read more. If your product name is not listed there, refer to your product manufacturer’s website.

2 To download a plug-in for your application, go to the Authoring Tools section of Viewpoint Developer Central. Click the name of your application and follow the on-screen instructions. Note: 3ds max, Maya, and LightWave plug-ins are listed under “Free Applications” near the bottom of the page.

### Before You Export or Save to Viewpoint Technology

For best results, use the feature set of your 3D authoring application to create the following for your 3D model:

- Geometry (mesh) and textures for your 3D object.
- Complex animations, including camera animations. Viewpoint also supports the export of morphs from Poser4 Pro Pack.

When your 3D model is complete, you can save or export to Viewpoint media files.

Lists of supported and unsupported features are available on Viewpoint Developer Central for several 3D modeling applications. Click a link below and review the "Supported Features" and "Unsupported Features" sections for your application:

- 3ds max
- Lightwave
- Maya
- Cinema 4D
- trueSpace
- RapidForm 2001
- Carrara

**Note:** Once exported to Viewpoint Media Files, you can adjust your scene in Viewpoint Scene Builder. However, much of your basic scene can and should be completed in the native 3D modeling application.
Quality Check 3D Scenes in the Native Applications

Follow these guidelines for checking the quality of a file before exporting from the native application:

- Set the base location (also called "world center") for your 3D scene to 0, 0, 0. Also check the pivot points.
- Optimize models to include the minimum number of polygons, while still maintaining visual quality.
- Check the quality of smoothing. Be sure to use smoothing groups whenever possible.
- Rotate the models to check for hidden objects such as placeholders.
- Make sure any image files used for textures are saved using a progressive JPEG file format with RGB color and with an image size that is a power of 2. Examples of proportions that are powers of 2 are 32 x 32, 64 x 256, and 1024 x 32.
- Make sure that text is not inverted.
- Zoom in to inspect the geometry.
- Make sure that none of the polygons have flipped. You can check this by applying a material to the mesh and then changing the material from two-sided to one-sided.
- Test all animations and make note of the order and frame ranges in which you want them to occur. When loading the .ase file later, use this information to organize your animations by specifying and naming groups of frames in the Scene Builder ASE Behavior Editor dialog box.
- Scale your scene. For details, see the following section, “About Scaling Scenes”.

About Scaling Scenes

Upon creating a scene in your 3D authoring application, it's important to scale the scene, especially if you plan to use object transformations or camera animations in your Viewpoint Technology application.

Specifically, the units in the scene should match the Viewpoint unit-based system of controlling scene parameters. Objects in a Viewpoint Technology application should have an X, Y, and Z scale of "1" or 100% and a position of 0, 0, 0 in the Viewpoint coordinate system. These settings allow you to quickly transform and animate elements in the scene and then be able to change them back, if necessary.

Also, if you import a scene with large dimensions in it into a Viewpoint scene, the typical camera orbit distance of 4 "units" will not suffice. For example, if you import an aircraft carrier modeled in real life dimensions into a Viewpoint scene, you'll need to change the orbit distance to a much larger number, perhaps 1000 units. At a certain point, manipulating these numbers becomes unwieldy and causes the scene to shoot off in space as soon as a user tries to navigate.

If you are using 3D Studio, this scaling action typically involves grouping the entire scene together and resizing it so that the entire scene fits into a 2" x 2" x 2" box, moving it to the 0, 0, 0 world coordinates, ungrouping it, and then exporting it. By following the basic steps below, you will minimize the chance of problems cropping up later, such as when the Viewpoint scene has been published.
To scale your scene

1. After modeling your scene with mapping coordinates and textures applied, collapse all your objects' modifier stacks down to Editable Meshes.

2. Group all your objects together using the group command.

3. Set your units setup to U.S. Standard (feet and inches.)

4. Create a box that is 2" x 2" x 2" in size and that is centered on the 0, 0, 0 coordinate.

5. While your objects are still grouped, use the Scale tool to uniformly scale your objects to fit into the box.

   **Important:** Remember to delete the box once finished.

6. Select your group and open the **Uniform Scale** dialog box. Notice that the X, Y, and Z scales are no longer 100%.

7. Use the **Reset X Form** command to reset these parameters to 100%.

8. Optionally, ungroup your objects.

   Keep in mind that if you choose to keep objects grouped, they will come into your Viewpoint Technology scene grouped within a parent instance. Typically, you will want to ungroup them before exporting.

You can verify that the above steps worked correctly by examining the X, Y, and Z Scale parameters in your .mtx file. They should all be set to 1. Similarly, the rotation and location parameters should be 0. If you want to resize your objects, try using the **Transform** function in Scene Builder. Because the objects have a scale of 1, you can also use the Scene Builder **Set Identity** command to bring them back to their correct scale quickly.

Exporting or Saving to Viewpoint Technology Media Files

Upon completing your 3D model, you can export or save your model to Viewpoint Technology format. The process for doing this depends on the application you are using. For more information, see “3D Modeling Applications That Support Viewpoint Technology”.

Many 3D modeling applications allow you to save or export directly to Viewpoint Media Files. For specific steps on how to save or export from your 3D modeling application, check the application documentation or online Help.

Specific authoring steps are available on Viewpoint Developer Central for several 3D modeling applications. Click a link below to see the authoring steps for your application:

- **3ds max**
- **Lightwave**
- **Maya**
- **Cinema 4D**
- **trueSpace**
- **RapidForm 2001**
- **Carrara**
Exporting from 3ds max, Maya, or LightWave

Viewpoint has released media exporter plug-ins for 3ds max, Maya, and LightWave. You can install these plug-ins free of charge from Viewpoint Developer Central. Click a link below to go to a page describing the plug-in for your application. From there, you can download the plug-in and related documentation:

- Viewpoint Media Exporter for 3ds max 3.1 and 4
- Viewpoint Media Exporter for Maya 3 and 4
- Viewpoint Media Exporter for LightWave 6.5

Use PolyTrans to Convert Files

If your 3D application does not support Viewpoint Media File export, you can probably convert your scene using PolyTrans. PolyTrans is a 3D scene translation tool allowing you to import and export many different 3D file formats.

For instructions on converting 3D scenes to Viewpoint Technology with the Viewpoint Technology Export Filter interface, go to the Okino website (http://www.okino.com/conv/exp_vet.htm). Click the screenshot for each export filter tab — Quality, Publish, Enable, Materials, and Animations — to find about its settings.

Placing a Viewpoint Technology Web Application Into a Web Page

Upon converting your 3D model to Viewpoint Technology, you can place it into a web page and test it. Most 3D authoring applications exporting to Viewpoint Technology let you publish your 3D model directly to a web page.

If your application doesn't include this export feature, you can use Viewpoint Media Publisher (available from Viewpoint Developer Central) to place your model into a web page.

To embed your 3D model into a web page using Media Publisher

1 Start Viewpoint Media Publisher.

2 From the HTML Template menu, choose generic.html.

3 Drag the .mtx file for your 3D model onto Media Publisher, or click the Select MTX/MTZ button and select the file.
4 Type a name for the .html file, and then click **Save**.

5 Double-click the icon for the newly created .html file to view it in your default web browser.

**Testing Your 3D Model's Animation**

Media Publisher automatically creates JavaScript buttons that you can use to trigger and test your Viewpoint Technology web application animations. This is assuming you have set On="0" for each animation. If you set On="1" for an animation, Viewpoint Media Publisher will not create a button for that animation.

**Important:** Before you add any other functionality, be sure to test all animations for your 3D scene to ensure that they perform correctly. If something doesn't work correctly, you can use your 3D modeling application to adjust your 3D model and then re-export it to Viewpoint Technology as described earlier.

For more information on testing animation, see “Testing Animations and Interactivity”.

**Adjustments You Can Make After Exporting to Viewpoint Technology**

Upon exporting a scene from your 3D modeling application, you can use Viewpoint Scene Builder and other tools to further refine to your scene.

You can use Viewpoint Scene Builder to improve your scene with the following:

- **Procedural lightmaps** — You can use the **Create Lightmaps** button on the **Materials/Textures** menu to create and customize lightmaps. A lightmap allows you to mimic a unique light source, such as diffused light, and is another way to make the
object in your scene appear more lifelike or to add a special effect. With Create Lightmap, you can set the direction of the light source, the color, specularity, and intensity. You can also add JPEG image information with little noticeable effect on file size, as compared to the standard method of applying a JPEG lightmap. For more information, see the Viewpoint Scene Builder User Guide available from Viewpoint Developer Central.

- Texture optimization — In Texture Lab, you can optimize the scene texture, add alpha map transparency, add a noisemap, and choose settings by channel. You can also control the type and amount of compression you want to apply. For more information, see the Viewpoint Scene Builder User Guide available from Viewpoint Developer Central.

- Animation controls — Divide animations by keyframe animations imported from other applications. This allows you to edit or add to the animations using the Viewpoint XML commands in the .mtx file. For more on adding animation, see Chapter 3: “Adding Animations to Your Viewpoint Technology Web Application”.

- Hot spots with rollover text — Hot spots are procedural geometric primitives that have a vector-based text string displayed by a mouse-over trigger by default. Use hot spots to identify parts of a scene for a user interacting with it. For more information, see Chapter 4: “Adding Hot Spots to a Viewpoint Technology Web Application”.

- SpecWrap — Creates a more realistic light source effect in a scene with multiple objects by adjusting the way light wraps around each object.

SpecWrap adjusts the tension of a lightmap when applied to a piece of geometry. As the SpecWrap value is increased, the lightmap parameters are stretched on the geometry, altering its appearance. The default value is SpecWrap=“0” and it has a range of 0 to 10. It can either be adjusted directly in the XML by adding SpecWrap=“X” to the MTSGeometry tag, or in Scene Builder it can be found on the Instances (2) menu. To adjust its value in Scene Builder, select the geometry with the applied lightmap and adjust the slider.

- Environmental Dithering — Removes banding artifacts (caused by lightmaps) that sometimes appear. Use the Dither slider on the Textures (6) menu in Scene Builder.

You can also make these adjustments to a scene after exporting from a 3D modeling application:

- Flash animations — You can create a Flash preloader file to play while the rest of your scene loads. You can also use Flash animations as foreground and/or background to your scene. For details, see Chapter 6: “Adding Flash to a Viewpoint Technology Web Application”.

- Add texture animations — You can apply a JPEG image or a Flash movie as a texture on an object or hot spot. For related information, see Chapter 6: “Adding Flash to a Viewpoint Technology Web Application” and Chapter 4: “Adding Hot Spots to a Viewpoint Technology Web Application”.

- Interactivity to trigger animations — You can define what interaction (for example, a mouse click on a certain hot spot) will cause an animation to play. For details, see Chapter 5: “Triggering Animations”.

For related information, visit the Examples & Tips section of Viewpoint Developer Central.
Chapter 3: Adding Animations to Your Viewpoint Technology Web Application

About Animations in a Viewpoint Technology Web Application

In most cases, you'll import complex animations from the 3D authoring tool in which you create your scene. You can also create simple animations by using the Viewpoint XML script in the .mtx file.

For detailed information about Viewpoint Technology XML, see the Viewpoint Technology XML Reference Guide available from the Viewpoint Developer Central.

Imported Animations

When importing animations from your 3D modeling application, be sure that the animations are split to match the parts of the animation you want to reference. For example, for an animation that both unfolds and rotates the lawn chair, you may want to split the animation into two: one to unfold the chair, one to rotate it. This creates separate triggers for the separate animations.

You can split animations using:

- Your 3D modeling application
- Viewpoint Media Exporter for 3ds max
- Viewpoint Scene Builder

Effective Ways to Use Simple Animations

You can use animations to highlight features of your product and to demonstrate operability. For example, with animation you could:

- Give a little spin — Show off the overall view of your product by rotating it 360 degrees. Then, let it rest in an interesting position so that your potential customer can take over the navigation controls.
- Open and close doors and hatches — Show what's inside by opening doors, unfurling banners, or decanting bottles.
- Walk it or roll it — If your 3D model has legs or wheels, show your potential customer how it moves.

**Animation Tip:** By default, animations run only once. Use the Clamp attribute to control the number of times your animation loops: Clamp="0" loops the animation infinitely; or specify a positive integer for the number of times you want the animation to loop. In this example, the animation loops three times:

```
<MTSTimeElem Type="Keyframe" Name="Turn" On="1" Clamp="3" />
...  </MTSTimeElem>
```
Creating Simple Animations

The following animations allow you to take advantage of the 3D capabilities of a scene. If you are unable to export animations from your 3D modeling application to Viewpoint Technology, you can easily create the following simple animations by editing the .mtx file for your Viewpoint Technology web application.

Scripting a Translation Animation

A translation animation moves an object from one x, y, z coordinate in the scene to another.

```xml
<MTSTimeElem Type="Keyframe" Name="ObjectTranslate" On="1">
  <Target Name="Simple_0" Property="loc_" Timeline="T0" />
  <Time>
    <Timeline Name="T0" Type="3D"> [-1.472 -1.500 0.000 ] [ 1.400 -0.500 0.000 ]</Timeline>
  </Time>
</MTSTimeElem>
```

The shaded items above indicate the values that are user dependent. Keep the rest of the code as it is.

The following steps demonstrate how to get object translation coordinates from Viewpoint Scene Builder. To find out how to determine object rotation coordinates in your 3D modeling application, consult its print or online documentation.

**To determine x, y, and z translation coordinates using Scene Builder**

1. Open Scene Builder.
2. Drag and drop your .mtx file onto the Scene window in Scene Builder. This opens the scene.
3. Click the instance name at the top of the Hierarchy menu on the left.

4. In a scene with only one model, click the instance name following MTSRootInstance. Or, in a scene with multiple models, right-click the object you want to select to select the instance name in the Hierarchy menu.
Scene information displays in the Object Info window at the bottom of your screen as shown in the image below.

5 Right-click and drag the object to translate it.

6 In the Hierarchy menu, click the name of the object to display Instance information in the Object Info window.

7 In the Object Info window, click the Translate line. This copies the coordinates to the clipboard.

8 Paste these into a text file to edit them for use in your .mtx file.

9 Upon gathering the coordinates you want, edit the coordinate text to conform to the format of Timeline values.

For example, this text from the Object Info window:

Translate x="-1.422" y="-1.572" z="0.000"

It should look like this as a Timeline value:

[-1.422 -1.572 0.000]

These numbers are the x, y, and z coordinates, respectively.

10 Include the 3D coordinates in the animation element (MTSTimeElem) for each Time value. For each of the two time values (0 and 5) there is a set of corresponding Timeline coordinates contained in brackets ([ -1.422 -1.572 0.000 ] [ 1.400 -0.500 0.000 ]).

```xml
<MTSTimeElem Type="Keyframe" Name="ObjectTranslate" On="1" >
  <Target Name="Simple_0" Property="loc_" Timeline="T0" />
  <Time> 0 5 </Time>
  <Timeline Name="T0" Type="3D"> [-1.422 -1.572 0.000] [1.400 -0.500 0.000] </Timeline>
</MTSTimeElem>
```
The shaded areas indicate values that are user-dependent. Change these to suit your Viewpoint Technology web application. Keep the rest of the code as it is.

**Note about On=:** If you set On="1" the animation runs when it is loaded. If you set On="0" the animation runs when it is triggered. Many examples in this guide use On="1", but for testing purposes, you may want to use On="0" so that you can manually trigger each animation. For more information, see "Testing Animations and Interactivity".

11 Upon completing the process of editing the animation, use Media Publisher to embed the scene in a web page. See “Placing a Viewpoint Technology Web Application Into a Web Page”.

12 Test the animation for your Viewpoint Technology web application. See “Testing Animations and Interactivity”.

### Scripting a Rotation Animation

A rotation animation provides a good way to show a single object in a scene. A rotation animation simply rotates the object while the rest of the scene remains still.

For steps on how to determine object rotation coordinates in your 3D modeling application, consult its print or online documentation. The following steps show you how to get object rotation coordinates from Viewpoint Scene Builder.

**To determine x, y, and z rotation coordinates using Scene Builder**

1. Open Scene Builder.

2. Drag and drop your .mtx file onto the Scene window in Scene Builder. This opens the scene.

3. Click the instance name at the top of the **Hierarchy** menu on the left.

4. In a scene with only one model, click the instance name following MTSRootInstance. Or, in a scene with multiple models, right-click the object you want to select, and the instance name is selected in the **Hierarchy** menu.
Scene information displays in the Object Info window at the bottom of your screen.

5 Right-click the object and drag it to rotate the object to the coordinates you want.

6 In the Hierarchy menu, click the name of the object to display Instance information in the Object Info window.

7 In the Object Info window, click the Rotate line. This copies the coordinates to the clipboard.

8 Paste the coordinates into a text file to edit them for use in your .mtx file.

9 Upon gathering the coordinates you want, edit the coordinate text to conform to the format of Timeline values. For example, this text from the Object Info window:

```
Rotation x="27.607819" y="-45.870430" z="-35.150375"
```

It should look like this as a Timeline value:

```
[27.607819 -45.870430 -35.150375]
```

These numbers are the x, y, and z coordinates, respectively.

10 Include the rotation coordinates in the animation element (MTSTimeElem) for each Time value. For each of the three time values (0, 5, and 10) there is a set of corresponding Timeline coordinates contained in brackets ([27.607819 -45.870430 -35.150375], [0 90.000 0], [90.000 0 0]).

```
</MTSInstance>
<MTSTimeElem Type="Keyframe" Name="ObjectRotate" On="1" >
  <Target Name="Simple_0" Property="rot_" Timeline="T0" />
  <Time> 0 5 10 </Time>
  <Timeline Name="T0" Type="3D" >
    [27.607819 -45.870430 -35.150375] [0 180.000 0] [0 360 0]
  </Timeline>
</MTSTimeElem>
```
Scripting a Camera Animation

Camera animations are a good way to demonstrate product features and change a visually static scene into a dynamic and interactive one. In a camera animation, you are scripting the camera to move around the center point of the scene. Objects within the scene remain stationary. This is in contrast to a rotation animation in which the object moves while the camera remains still.

**Tips for Camera Animation:** Set clockwise camera rotation by using positive Timeline values from 0 through 360 (degrees). Use negative numbers in the Timeline values to change the direction of a rotation; for example:

```xml
<Timeline Name="T0" Type="3D"> *
  [-1.765 -1.876 0] [0 -90.000 0] </Timeline>
```

For steps on how to determine camera coordinates in your 3D modeling application, consult its print or online documentation. The following steps show you how to get camera coordinates from Viewpoint Scene Builder.

**To determine x, y, and z camera coordinates using Scene Builder**

1. Open Scene Builder.
2. Drag and drop your .mtx file onto the Scene window. This opens the scene.
3. Click Camera at the top of the Hierarchy menu on the left. The camera information displays in the Object Info window at the bottom of your screen.
4. Click the object and drag to rotate the camera to the coordinates you want.
5. In the Object Info window, click the Rotate line to copy coordinates to the clipboard. Paste these into a text file, so that you can edit them for use in your .mtx file.
Upon gathering the coordinates you want, edit the coordinate text to conform to the format of Timeline values.

For example, this text from the Object Info window:

```
Rotate x="25.829" y="-49.822" z="0.000"
```

It should look like this as a Timeline value:

```
[25.829 -49.822 0.000]
```

These numbers are the x, y, and z coordinates, respectively.

Include the camera coordinates in the animation element (MTSTimeElem) for each Time value. For each of the three time values (0, 5, and 10) there is a set of corresponding Timeline coordinates contained in brackets (*, [25.829 -49.822 0.000], [0 90.000 0]).

```
<MTSTimeElem Type="Keyframe" Name="RotateCamera" On="1" >
<Target Name="MTSCamera" Property="crot" Timeline="T0" />
<Time> 0 5 10 </Time>
<Timeline Name="T0" Type="3D" > * [25.829 -49.822 0.000] [0 90.000 0]
</Timeline>
</MTSTimeElem>
```

**Note:** The asterisk in the Timeline is a wildcard value, meaning that the first camera coordinates are whatever the camera position is at the time the animation begins running.

### Creating Texture Animations

Texture animations allow you to show a model with different texture options. This is a great way to demonstrate a product in different colors or fabrics, for instance. The following sections describe how to animate textures from JPEG files that are externally referenced by the published Viewpoint files.

### Creating Your Own Files for Textures

You can use any image-editing software (such as Adobe Photoshop) to create your textures.

Be sure to keep in mind these restrictions:

- Use a progressive JPEG file format for all texture image files.
- Make sure to save the image using RGB color format.
- For best results, keep the image size to a power of 2. (Examples of proportions that are powers of 2 are 32 x 32, 64 x 256, and 1024 x 32.)

**Note:** You can also apply a Flash movie as the texture for an object. For example, on a 3D model of an electronic device with a display screen, you could use a Flash movie to show what the display looks like. For more information on how to apply a Flash movie as a texture, see the Using Vector Graphics in Viewpoint Technology Guide available from Viewpoint Developer Central.
Editing the .mtx File to Animate a Texture

The process of animating textures includes four parts:

- Declare textures for animation
- Set the textures to preload
- Create texture animators
- Script a trigger for the animation

This chapter describes the first three steps in this process. The final step, triggers, is described in Chapter 5: "Triggering Animations".

To declare textures for animation

1. Find the name of the texture that is currently assigned to the object for which you want to animate textures. This name, found in the MTSTextureMap tag (as highlighted in the following example), is your target texture.

   For example, in the following, the texture is named "cube_TEXTURE_0".

   ```xml
   <MTSInstance Name="some_name_MESH_0">
     <MTSGeometry Name="some_name_MESH_0" BackFaceDir="0"
 CreaseAngle="60.00001" />
     <MTSMaterial Name="some_name_MATERIAL_0" ID="0">
       <MTSTextureMap Type="Diffuse" Name="cube_TEXTURE_0" />
       <MTSTextureMap Type="Light" Name="some_name_TEXTURE_1" />
     </MTSMaterial>
   </MTSInstance>
   ``

2. Add the texture declaration before the first MTSInstance tag in the .mtx file.

   ```xml
   <!-- Texture Declaration -->
   <MTSTextureMap Name="cube_TEXTURE_0_image1" />
   <MTSTextureMap Name="cube_TEXTURE_0_image2" />
   <MTSTextureMap Name="cube_TEXTURE_0_image3" />
   <!-- Preloading Textures -->
   <MTSTimeElem Type="MTSImageStream" On="1" Path="Textures/circle.jpg">
     <Target Name="MTSTexture.cube_TEXTURE_0_image1" />
   </MTSTimeElem>
   <MTSTimeElem Type="MTSImageStream" On="1" Path="Textures/logo.jpg">
     <Target Name="MTSTexture.cube_TEXTURE_0_image2" />
   </MTSTimeElem>
   ``

Now you are ready to set textures to preload. Preloading textures into your scene is important to ensure a smooth dissolve between the animating images.

To set textures to preload

1. Identify the names of the files you will for each of the textures. This example uses three external files in the Textures directory called "circle.jpg", "logo.jpg", and "viewpoint.jpg". These will be used for cube_TEXTURE_0_image1, cube_TEXTURE_0_image2, and cube_TEXTURE_0_image3 respectively.

2. Add the following code into the .mtx file before the closing </MTSScene> tag.

   ```xml
   <!-- Preloading Textures -->
   <MTSTimeElem Type="MTSImageStream" On="1" Path="Textures/circle.jpg">
     <Target Name="MTSTexture.cube_TEXTURE_0_image1" />
   </MTSTimeElem>
   <MTSTimeElem Type="MTSImageStream" On="1" Path="Textures/logo.jpg">
     <Target Name="MTSTexture.cube_TEXTURE_0_image2" />
   </MTSTimeElem>
   ```
Once you've set textures to be preloaded, you can create texture animators. In most cases, you'll need to create two animations to change texture on the object:

- One animation to cycle forward through the three textures
- Another to cycle back to the original texture

**To create a texture animator**

1. Below is the MTSTimeElem used to preload textures, add the script for the first texture animation.

```xml
<MTSTimeElem Name="TextureAnimation" Type="Keyframe" On="0">
  <Target Name="cube_TEXTURE_0" Property="pixl" Timeline="T1" />
  <Time> 0 1 3 </Time>
  <Timeline Name="T1" Type="Texture">
    [MTSTexture.cube_TEXTURE_0_image1]
    [MTSTexture.cube_TEXTURE_0_image2]
    [MTSTexture.cube_TEXTURE_0_image3]
  </Timeline>
</MTSTimeElem>
```

**Note:** The Time tag indicates time in seconds along a timeline. The Timeline tag then describes what will happen at each point in the Time. So at Time0, the first texture is animated onto the object, at Time1 the next texture, and at Time2, the final texture.

2. Below the first texture animation, add the script for the reverse texture animation.

```xml
<MTSTimeElem Name="TextureAnimationReverse" Type="Keyframe" On="0">
  <Target Name="cube_TEXTURE_0" Property="pixl" Timeline="T1" />
  <Time> 0 1 3 </Time>
  <Timeline Name="T1" Type="Texture">
    [MTSTexture.texture2_image3]
    [MTSTexture.texture2_image2]
    [MTSTexture.texture2_image1]
  </Timeline>
</MTSTimeElem>
```

Now you are ready to create a trigger for the texture animation.

As described in Chapter 5: “Triggering Animations”, you can add triggers to either the scene . mtx file or to the web page.

To find out more about creating interactivity in Viewpoint Technology web applications, see the Creating Interactive Viewpoint Technology Content Guide available from Viewpoint Developer Central.
Chapter 4: Adding Hot Spots to a Viewpoint Technology Web Application

What is a Hot Spot?

Sometimes called "widgets," hot spots are definable 3D areas that let you add interactivity and actions to a Viewpoint Technology web application. You can either define the action to occur when a user points or clicks each hot spot, or you can define a hot spot to always display what you associate with it.

You can use a hot spot to display:

- A text string
- A 2D image (JPEG file)
- A Flash movie

You can use hot spots to show information about an object or its features, or to trigger an animation or some other action. For information on creating and using triggers, see Chapter 5: "Triggering Animations".

Creating Hot Spots With Rollover Text

You can make your Viewpoint Technology application more dynamic and interactive by adding one or more hot spots with rollover text. Rollover text (text that pops up when the mouse pointer travels over a specified area in the scene) is a powerful way to communicate with your viewers, allowing you to educate or direct them.

In the graphic below, the hot spot (shown as a red sphere) is visible to illustrate how hot spots are defined. Usually the hot spot itself is invisible.

In this case, when the mouse pointer passes over this hot spot, text about the footstool adjustment mechanism displays.

© 2003 Viewpoint Corporation. All rights reserved.
To add a hot spot using Scene Builder

1. Open Scene Builder.

2. On the Hot Spot menu (7), click Add Hot Spot and choose the hot spot shape you want to add: sphere, cube, cylinder, cone, pyramid, or plane.

3. Click the Show Hot Spots and Show Hot Spot Text menu options to set each to On. This allows you to see the hot spots while you are setting the related options. You can set both of these options to Off before you save your file.
   - Show Hot Spot Text — When On, hot spot text is always visible, regardless of whether the user points to it with the mouse pointer.
   - Show Hot Spots — When On, the hot spot shape is visible. Generally, this is set to Off in a scene.

4. Click Hot Spots Options to set it to On.

5. Move the hot spot to where you want the rollover to be.

6. From the Transformation menu (4), use the Scale X, Y, and Z sliders (to scale each vector separately) or Uniform Scale sliders to resize the hot spot to the dimensions you want.
On the Hot Spot (7) menu, click Rollover Text, and then type the rollover text you want. Click Font or Text Color to change the font style and color.

Tip: If you want to wrap lines of rollover text, use \n new line escape sequences to break up the lines. For example,

"My word, this is a long, long text string!"
can be broken up with \n
"My word, this is \n a long, long \n text string!"
to create three lines of text.

To learn more about advanced editing of hot spot such as triggering animation or other action, or applying textures, see the Viewpoint XML Authoring Guide and experiment with the Hot Spot menu in the Scene Builder.

Creating a Hot Spot to Display a 2D Graphic

To create a hot spot with a texture in Scene Builder

1 Open Scene Builder.

2 On the Hot Spot (7) menu, click Add Hot Spot and then click the shape you want to use for this hot spot.

3 Click Show Hot Spot Text to change this option to On.

4 Click Displayable Texture and click Load.

5 Select any JPEG file as the texture for this object.

6 From the Publish (0) menu, click Save MTX to save the .mtx file for this scene.
Using a Hot Spot to Display a Flash Movie

Perform the following steps to apply a Flash movie as a texture on hot spot

1. Open your .mtx file in any text or XML editor. For example, suppose the .mtx file for your scene includes only one hot spot with a JPEG texture, as in the following:

   `<MTSScene Version="4" >
   <MTSSceneParms RenderMode="LightTexture" BlendShadow="1"
   EdgeBias="1" />`  
   `<MTSCamera OrbitDist="4" >
   <ViewLocation x="0"  y="0"  z="4" />`
   `</MTSCamera>`  
   `</MTSScene>`

2. Near the end of the file, find an MTSTimeElem element that loads the JPEG file you applied in Scene Builder. For example:

   `<MTSTimeElem Type="MTSImageStream" On="1"
   Path="images/candle.jpg" >`
   `</MTSTimeElem>`

3. Give the MTSTimeElem element a name and replace the Type and Path values as shown in the following example:

   `<MTSTimeElem Name="myflash" Type="SWFView" On="1"
   Path="flashfile.swf" >`
   `</MTSTimeElem>`

4. Following the MTSCamera tags, find an MTSTextureMap tag like the following:

   `<MTSTextureMap Name="candle" />`

5. Add X and Y values to match the dimensions of your Flash movie. Be sure to use coordinates that are powers of 2.

**Note:** The texture name used in this file was automatically derived from the JPEG file you used originally. If you want to change this texture name, be sure to change it consistently throughout the .mtx file. However, because the texture name is simply a tag, changing it is not necessary—it does not need to match the name of your .swf file.
Also add HasAlpha tag. Typically, you will set HasAlpha="1". For more information about HasAlpha and a related tag, Comp, see the Viewpoint XML Reference Guide, available from Viewpoint Developer Central.

The result will be a line similar to the following:

```
<MTSTextureMap Name='candle' x='128' y='128' HasAlpha='1' />
```

**Note:** It's important to use textures sized to powers of two such as 128 x 128, 256 x 64, 512 x 128, etc.

Because Flash files are vector-based, the X and Y coordinates don't have to match the texture height and width. You can save space without losing fidelity by setting a Flash texture to a smaller dimensions (E.g., 64 x 64) and then scaling it to a larger size (E.g., 512 x 512).

6 Save the changes to your .mtx file.

**Pinning a Hot Spot Image or Flash Animation**

You can pin a hot spot text, graphic, or Flash animation to a fixed location by specifying the source and destination coordinates in the .mtx file. Pinning means that you are designating a coordinate spot in the scene where a texture applied to a hot spot will always be located. This ensures that the hot spot text, graphic, or Flash animation always appears in the same location, regardless of the angle from which the scene is being viewed.

In the .mtx file, you can specify source and destination coordinates in pixels or as a percentage of the Viewpoint Technology application window.

Destination Pin Flag “0”

Source Pin Flag “0”

This side sets the relative flags to “1”, which expresses the pin locations as a percentage.

- `SrcPinRel="1"`
- `DstPinRel="1"`

The Source and Destination Pin settings are, in this example:

- `SrcPin="-0.5 -0.3"`
- `DstPin="-0.3 -0.3"`

Texture 64 x 128 Pixels

-30%

-50%

-30%

VET Web Application Window 300 x 300

© 2003 Viewpoint Corporation. All rights reserved.
The coordinate system allows developers to select any of the four corners of a texture/text or the scene window as the origin point for pinning hot spots. Relative pinning allows the pin location to be specified as a percentage relative to the origin. So, for example, if the Viewpoint Technology application switches to HyperView mode, the hot spots maintain their same positions.

The illustration above shows how the SrcPin, DstPin, SrcPinRel, DstPinRel, SrcPinFlag, and DstPinFlag tags are used.

**Note:** The SrcPinRel and DstPinRel tags are supported as of Viewpoint Technology release 3.0.8. For more information, see the Viewpoint XML Reference Guide.

**To pin a hot spot’s text, graphic, or Flash movie to a fixed location**

1. Using the text or XML editor of your choice, open the .mtx file.

2. In the LayerData element for the texture (text, graphic, or flash movie) you want to pin, add these attributes:
   
   - **SrcPinRel** to set the SrcPin value to pixels (SrcPinRel="0") or as a percentage (SrcPinRel="1").
   
   - **DstPinRel** to set the DstPin value to pixels (DstPinRel="0") or as a percentage (DstPinRel="1").
   
   - **SrcPin** to specify the X and Y offsets for the source of the pinned hot spot.
   
   - **DstPin** to specify the X and Y offsets for the destination of the pinned hot spot.
   
   - **SrcPinFlag** and **DstPinFlag** to specify which corner of the source and destination rectangles to pin (0=upper-left, 1=upper-right, 2=lower-left, 3=lower-right). It's best to make this value match the value for **SrcPinFlag**.

   This is an example:

   ```xml
   <LayerData Name="some_name"  Text="Rollover text goes here." TextColor="255 213 44"  Language="English"  TextSize="32" Center="0.000 0.000 0.000"  TrackPoint="0.000 0.000 0.000"  Radius="6" Shadow="1"  Anchor="0.000 0.000"  SrcPinRel="3"  DstPinRel="3"  SrcPinFlag="1"  DstPinFlag="1"  SrcPinRel="3"  DstPinRel="3"  SrcPin="-0.5 -0.3"  DstPin="-0.3 -0.3"  AnchorWidget="0"  WidgetClamp="0"  WidgetLine="1"  AlwaysVisible="0"/>
   ```
Setting the Cursor to Indicate Clickable Hot Spots

For hot spots that set to display information with a mouse click, you can set the cursor to cue the user.

**To set the cursor to indicate hot spots**

1. In Scene Builder, add your hot spots. For steps on creating hotspots, see “Creating Hot Spots With Rollover Text” and “Creating a Hot Spot to Display a 2D Graphic”.

Following is an example of the XML code for a hot spot. This code relates to the graphic above.

```xml
<MTSInstance Name="Simple_0" Visible="0" DoShadow="0">
  <Transform>
    <Scale x="0.5" y="0.5" z="0.5" />
  </Transform>
  <MTSGeometry Name="MTSSimple_0" Type="MTSSphere" Resolution="10" />
  <MTSInstance Name="Layer2D_0" DoShadow="0" Collapsed="1">
    <Transform>
      <Scale x="0.2" y="0.2" z="0.2" />
    </Transform>
    <LayerData Text="Rollover Text" TextColor="255 213 44" />
  </MTSInstance>
</MTSInstance>
```

In this case, "Simple_0" is the name of the clickable hot spot. You can use any instance name for your hot spot by replacing "Simple_0" with that name.

2. Save the .mtx file.

**Important**: To avoid degradation of quality, do not republish; use the Save option instead.
3 Use an XML editor to edit the .mtx file. Below the MTSInstance element for the hot spot, add this code:

```xml
<MTSInteractor Name="cursors" >
  <Target Name="MTSInstance.Simple_0"/>
  <OnEnter Action="MTSAssignProperty" Target="MTSScene::crsr" Value="MTSCursor::MTS_Point"/>
  <OnExit Action="MTSAssignProperty" Target="MTSScene::crsr" Value="0"/>
</MTSInteractor>
```

In this case, "MTSInstance.Simple_0" refers to the clickable hot spot named in step 1. If you renamed "Simple_0" in step 1, remember to rename it here also.

This code changes the cursor to a pointing hand when the user points to the "hot" area (the instance of geometry specified in step 1). It reverts to the default system cursor when the pointer leaves the hot area.

This hot spot also displays popup text ("Rollover Text") when the user points to the blue sphere.

### Creating a Hot Spot to Display Descriptive Text

You can use a hot spot to display narrative or descriptive text about various objects within your Viewpoint Technology application.

For example, suppose you had a Viewpoint Technology application with three objects, like the very simple website shown below. You could set up a hot spot to display text about each object as you point to it.

![Image of a simple website with a blue sphere and other objects]

**To create a hot spot to display descriptive text**

1. In the .mtx file, create a LayerData element to handle the text change. Use XML code like the following:

```xml
<MTSInstance Name="Layer2D_15" RenderLayer="1" Collapsed="1" >
  <LayerData Name="instructions" AlwaysVisible="0" Text="text" SrcPin="-15.000 -15.000" DstPin="0.000 0.000" AnchorWidget="1" WidgetLine="0" WidgetClamp="1"/>
</MTSInstance>
```

In this example, a placeholder for the text to be displayed is set by the Text attribute. The text change is handled by setting the text property in the LayerData element.

2. Add an MTSAction element to control when to display text on the hot spot texture. For example:

```xml
<MTSAction Name="text_handler"/>
```
This code does three things

- Sets the text property and accepts variables for the target (in this case the Layer2D_15 widget that holds the LayerData element) and the value or text string.
- Changes the cursor when the user mouses over the object and accepts variables for the type of cursor that will be shown.
- Controls whether the text is visible by collapsing or uncollapsing the hot spot.

For each object in your Viewpoint Technology application that you want to describe with text, create an interactor like the following:

```xml
<MTSInteractor Name="sphere_text">
  <Target Name="MTSInstance.Sphere" />
  <OnEnter Action="text_handler" txtatom="Layer2D_15" out="0"
    cursor="MTSCursor::MTS_Point" txttarget="instructions"
    txtval="Sphere
a procedural object created
in the XML"/>
  <OnExit Action="text_handler" txtatom="Layer2D_15" out="1"
    cursor="0"/>
</MTSInteractor>
```

This code implements the interactivity that will trigger the text change. The Target element specifies the object (in this case, "Sphere") that triggers the text change.

This code also includes two event handlers:

- The first is used when the user points to over the Target. It uncollapses the hot spot text, changes the cursor to be a pointer, and passes a text string to the "text" property in the LayerData element.
- The second is used when the pointer leaves the Target. It changes the cursor back to its default and collapses the text.
Chapter 5: Triggering Animations

Setting a Hot Spot to Trigger an Animation

You can create a 3D web application that allows you to trigger an animation when a user clicks or points to a hot spot. For example, when a user clicks a car door, it may cause the door to open. When the user points to the hood, it may become semi-transparent so that the user can see the engine.

To set a hot spot to trigger an animation

1. Use Scene Builder to create your hot spot. For details, see Chapter 4: “Adding Hot Spots to a Viewpoint Technology Web Application”.

   Important: To avoid degradation of quality, do not republish; use the Save option instead.

   Following is an example of the XML code for a hot spot:

   ```xml
   <MTSInstance Name="Simple_0" Visible="0" DoShadow="0">
     <Transform>
       <Scale x="0.5" y="0.5" z="0.5" />
     </Transform>
     <MTSGeometry Name="MTSSimple_0" Type="MTSSphere" Resolution="10" />
   </MTSInstance>
   <MTSInstance Name="Layer2D_0" DoShadow="0" Collapsed="1">
     <Transform>
       <Scale x="0.2" y="0.2" z="0.2" />
     </Transform>
     <LayerData Text="Click to open the car door" TextColor="255 213 44" />
   </MTSInstance>
   </MTSInstance>
   
   2. Create or identify the animation you want to trigger. This should be located at the end of the .mtx file before the closing </MTSScene> tag.

   For example:

   ```xml
   <MTSTimeElem Name="Animation_1" Type="Keyframe" On="0">
     <Target Name="MTSInstance.CarDoor" Property="loc_" Timeline="T1">
       <Time>0 2 5</Time>
     </Timeline>
     <Timeline Name="T1" Type="3D">* [0 -1 0] [0 1 0] </Timeline>
   </MTSTimeElem>
   
   This animation (called "Animation_1") changes the location of a geometry called "CarDoor".

   3. In the .mtx file, create a trigger for the animation. Place this before the animation.

   For example:

   ```xml
   <MTSInteractor Name="Anim_control">
     <Target Name="MTSInstance.Simple_0"/>
     <OnClick Action="Trigger" Target="MTSTimeElem.Animation_1"/>
   </MTSInteractor>
   
   This trigger causes the animation (animation_1) to occur when the user clicks the hot spot called "Simple_0".
Triggering an Animation From a Scene Object

You can enable users to click particular scene objects to trigger animation. For example, in the Viewpoint Technology web application shown below, when the user clicks the power (on/off) button, the front and side lights appear to illuminate.

To trigger animation from a scene object

1. In an XML editor, identify the object you want to use as the trigger. For example, in the Viewpoint Technology web application illustrated above, the power button, called "headlight_MESH_3" in the XML code is used as the trigger.

   For example, this is the code used to create the power button on the 3D model of the bike headlight:

   ```xml
   <MTSInstance Name="headlight_MESH_3" DoShadow="0">
     <MTSInstance Name="Layer2D_3" DoShadow="0" Collapsed="1">
       <Transform>
         <Scale x="0.2" y="0.2" z="0.2" />
         <Position x="340.169830" y="538.257324" z="-26.024282" />
       </Transform>
       <LayerData Text="on/off switch" TextColor="255 53 53" Font="Arial" TextSize="20" Radius="800" ShadowRadius="1" />
     </MTSInstance>
   </MTSInstance>
   
   2. Create or identify the animation you want to trigger. This should be located at the end of the .mtx file before the closing </MTSScene> tag.

   In this case, two animations ("anim_1" and "anim_2") are triggered by successive mouse clicks:

   ```xml
   <MTSTimeElem Name="anim_1" Type="Keyframe" On="0">
     <Target Name="headlight_TEXTURE_10" Property="pixl" Timeline="T1" />
     <Target Name="headlight_TEXTURE_7" Property="pixl" Timeline="T1" />
     <Time> 0 1 </Time>
     <Timeline Name="T1" Type="Texture">*[headlight_TEXTURE_13]
   </MTSTimeElem>
   
   <MTSTimeElem Name="anim_2" Type="Keyframe" On="0">
     <Target Name="headlight_TEXTURE_10" Property="pixl" Timeline="T1" />
     <Target Name="headlight_TEXTURE_7" Property="pixl" Timeline="T1" />
     <Time> 0 3 </Time>
     <Timeline Name="T1" Type="Texture">*[headlight_TEXTURE_3]
   </MTSTimeElem>
   
   3. In the .mtx file, create a trigger for the animation. Place this in the MTSInstance element for the object.
   ```
Triggerring Animations From HTML Using JavaScript

You can also trigger an animation from your HTML page using JavaScript. For example, in the website shown below, the **click to change color** button is coded with JavaScript to change the texture of the Compaq iPAQ™ Pocket PC in the Viewpoint Technology scene.

When a user clicks this button, the texture on the 3D model changes color.

To trigger animation from a scene object

1. Using an HTML editor, identify the object you want to use as the trigger.

   For example, in this case, the "buttons" are actually part of a 2D graphic called interaction_sample2.gif. Their locations are identified by their coordinates:

   ```html
   <img src="images/interaction_sample2.gif" width="198" height="157" border="0" usemap="#interaction"><br>
   <map name="interaction">
   <area shape="rect" coords="4,27,62,80"
   <area shape="rect" coords="70,27,123,80"
   <area shape="rect" coords="131,27,185,80"
   <area shape="rect" coords="29,90,82,143"
   </map>
   ```

2. Create or identify the animation you want to trigger. This should be located at the end of the .mtx file before the closing </MTSScene> tag.

   In this case, each button will trigger a separate animation. For example, this animation (called "animator_blue") will change the texture of the 3D model to blue:
3 Use JavaScript in the HTML file to create a trigger for the animation.

For example, the case below uses **onClick** commands to trigger animators defined in the .mtx file:

```html
<map name="interaction">
    <area shape="rect" coords="4,27,62,80" href="javascript:void(0)"
         onClick="vmp.TriggerAnim('animator_green')">
    <area shape="rect" coords="70,27,123,80" href="javascript:void(0)"
         onClick="vmp.TriggerAnim('animator_blue')">
    <area shape="rect" coords="131,27,185,80" href="javascript:void(0)"
         onClick="vmp.TriggerAnim('animator_red')">
    <area shape="rect" coords="29,90,82,143" href="javascript:void(0)"
         onClick="vmp.TriggerAnim('animator_black')">
</map>
```
Chapter 6: Adding Flash to a Viewpoint Technology Web Application

Flash Movies in Viewpoint Technology Web Applications

Flash .swf files—which contain scripted vector graphics, images, and text—can be played by both the traditional Flash Player and Viewpoint Media Player.

The difference is that Viewpoint Media Player can combine Flash animations with other rich media content, creating dramatic new effects in interactive content.

Flash files are much smaller than their pixel-based counterparts (such as .bmp or .gif files). While pixel-based files lose data and get blurry during compression, vector-based graphics lose no quality as they’re compressed.

In the web page below, the laptop is a 3D model. Here, Flash movies are used as a texture on the computer screen, as background and foreground effects, and as the texture on a hot spot (sometimes called a widget).

As described in Chapter 4: “Adding Hot Spots to a Viewpoint Technology Web Application”, you can use Flash movies to display when a user points or clicks on a hot spot. You can also add Flash movies as follows:

- **Foreground** — You can play a Flash movie in front of the other objects in your Viewpoint Technology web application scene.
- **Background** — Use a Flash movie as a background to your scene.
- **Texture** — Simulate movement or project some other animation on an object’s surface.
- **Preloader scene** — A preloader is a scene that plays while the main part of your Viewpoint Technology web application is loading. When it detects that the main scene has loaded, it launches the main Viewpoint Technology web application automatically.

This chapter describes how to add a Flash movie for each of these cases. For further details about using Flash files in your Viewpoint Technology application, see the Using Vector.
Adding a Flash Movie to Your Viewpoint Technology Web Application

Upon creating a Flash movie, all that is required to play a Flash movie in a Viewpoint Technology-enabled scene is to add an MTSTimeElem element to the .mtx file. Within that element, you can specify how the animation is rendered and when it is displayed.

**To add a Flash movie to your Viewpoint Technology web application**

1. Add an `<MTSTimeElem>` using the following syntax to the .mtx file:

   ```xml
   <MTSTimeElem Name="some_name" Type="SWFView" On="1"
   PreAnimator="1" Path="flashmoviefile.swf">
   <Target Name="MTSTexture.texturename" />
   </MTSTimeElem>
   ```

2. Modify the syntax of this element depending on how and when you want the Flash movie to display. See the following sections for details.

Using a Flash Movie in the Foreground or Background

You can use the PreAnimator and PostAnimator tags to place Flash movies in the foreground or background of a Viewpoint Technology scene.

This illustration below shows Flash movies playing in the background and the foreground.

PreAnimator and PostAnimator tags dictate when Viewpoint Media Player renders the animation in the render pipe. The rendering tree looks like this:

1. Render PreAnimator
2. Render 3D
3. Render PostAnimator

Use a PreAnimator or PostAnimator attribute to determine whether the movie plays in front (PostAnimator) of any 3D objects or behind (PreAnimator) of any 3D objects.

**Note:** You can use either PreAnimator or PostAnimator, but not both, for any single instance of a Flash movie.
To play the movie behind any 3D objects (in the background)

1 Include a PreAnimator attribute as following in your MTSTimeElem tag:

```
PreAnimator="1"
```

2 Use this syntax to place the movie in the scene:

```
<Target Name="MTSTexture.BackBuffer" />
```

The resulting code will look similar to the following:

```
<MTSTimeElem Name="background" Type="SWFView" On="1"
PreAnimator="1" Path="launch.swf" >
<Target Name="MTSTexture.BackBuffer" />
</MTSTimeElem>
```

To play the movie in front of any 3D objects (in the foreground)

1 Include a PostAnimator attribute as following in your MTSTimeElem tag:

```
PostAnimator="1"
```

2 Use this syntax to place the movie in the scene:

```
<Target Name="MTSTexture.BackBuffer" />
```

The resulting code will look similar to the following:

```
<MTSTimeElem Name="foreground" Type="SWFView" On="1"
PostAnimator="1" Path="plateAPrice.swf" >
<Target Name="MTSTexture.BackBuffer" />
</MTSTimeElem>
```

Using a Flash Image or Movie as a Texture

You can apply a Flash movie as a texture to any object in your Viewpoint Technology-enabled scene. In a Viewpoint Technology-enabled scene, Flash movie (.swf) files are externally referenced and replace existing textures in the .mtx file.

About Textures and UV Coordinates

While 3D objects are described in terms of their X, Y, and Z coordinates, the 2D textures applied to these objects are described in terms of U and V coordinates.

For example, if you had a 3D object of a bottle and wanted to apply a Flash image depicting its label, you would specify the UV coordinates of the rectangular area on which the 2D image will be mapped.

Textures are streamed into the Viewpoint Technology-enabled scene, so it's best to keep to a minimum the size of the target texture that is replaced by the Flash movie. The way you do this depends on several factors, the most important of which is whether your texture map requires explicit UV coordinates.
UV coordinates are required when the texture map does not cover the entire face of the object. An example might be a logo on the side of a product. The logo would be a separate texture map, as text typically requires a higher quality image (lower compression) so that the detail is not lost.

UV coordinates are not required when the texture map covers the entire object. However, you must know the exact size (pixels) of the texture map that covers the entire surface of the object, so you can set the aspect ratio for the Flash parent movie window. In this case, you can use a dummy texture declaration. A dummy texture is one that is given a name in a declaration statement but never equated with an actual file.

Finding the Name of An Existing Texture

You can use Viewpoint Scene Builder to identify the name of an existing texture.

To find a texture name using Scene Builder

1. Open your .mtx file in Scene Builder, and use the Hierarchy menu (upper-left corner of Scene Builder) to identify the name of the texture you want to replace with a Flash movie.

2. Make a note of the name of the texture. Quit Scene Builder without saving.

Note: In Scene Builder (3.0.7.x or later), you can double-click the texture name and type a new one. This allows you to make shorter, more descriptive texture names.
Applying a Flash Movie as a Texture

The process of applying Flash as a texture to a hot spot includes two parts:

• Defining the texture size and name
• Replacing the JPEG textures with Flash textures

The following sections describe the steps included in each of these parts.

Part 1: Define Objects, Hot Spots, and Their Textures In Scene Builder

Let Viewpoint Scene Builder do the work of creating the framework for your Viewpoint Technology-enabled scene. Create a scene that includes all the objects and hot spots that you have planned and position them where you would like them. For more information about using Scene Builder, see the Viewpoint Scene Builder User Guide which is available from Viewpoint Developer Central at http://developer.viewpoint.com/.

To create an object with a texture in Scene Builder

1. From the Scene Builder Primitives (8) menu, click the type of object you want to use.

2. Right-click the object to select it. Then from the Materials/Textures (6) menu, click Texture File and choose Load.

3. Select any JPEG file as the texture for this object.
4 From the Publish (0) menu, click Save MTX to save the .mtx file for this scene.

<table>
<thead>
<tr>
<th>Save MTX...</th>
<th>Publish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save MTX...</td>
<td>Publish</td>
</tr>
<tr>
<td>Compress File</td>
<td>Off</td>
</tr>
<tr>
<td>Verbose MTX</td>
<td>Off</td>
</tr>
<tr>
<td>Anim. in 3.0.5 Format</td>
<td>Off</td>
</tr>
<tr>
<td>Use Resource Names</td>
<td>On</td>
</tr>
<tr>
<td>Language Encoding</td>
<td>Off</td>
</tr>
<tr>
<td>Save MTX...</td>
<td>Publish</td>
</tr>
<tr>
<td>Image Quality</td>
<td>60</td>
</tr>
<tr>
<td>Lightmap Quality</td>
<td>20</td>
</tr>
<tr>
<td>Geometry Setting</td>
<td>0.8</td>
</tr>
<tr>
<td>Use TriXelsMT</td>
<td>On</td>
</tr>
<tr>
<td>Minimum Triangles</td>
<td>0</td>
</tr>
<tr>
<td>Retain Normals</td>
<td>Off</td>
</tr>
<tr>
<td>Publish...</td>
<td>Publish for Tuning Studio...</td>
</tr>
</tbody>
</table>

Note that when choosing Save MTX, the .mtx file retains an external reference to the JPEG file. By contrast, when choosing Publish, the JPEG file is embedded into the .mts file and the original JPEG file is no longer used.

**Part 2: Replacing JPEG With Flash Textures**

You are now ready to edit the .mtx file to replace the temporary JPEG textures you applied with Flash movie files.

**To apply a Flash movie as a texture on an object**

1 Open your .mtx file in any text or XML editor. For example, suppose the .mtx file for your scene includes only one hot spot with a JPEG texture, as in the following:

```xml
<MTSScene Version="4" >
  <MTSSceneParms RenderMode="LightTexture" BlendShadow="1"
  EdgeBias="1" />
  <MTSCamera OrbitDist="4">
    <ViewLocation x="0" y="0" z="4" />
  </MTSCamera>
  <MTSInstance Name="Simple_0">
    <Transform>
      <Scale x="0.5" y="0.5" z="0.5" />
    </Transform>
    <MTSGeometry Name="MTSSimple_0" Type="MTSCube" />
    <MTSMaterial ID="0">
      <MTSTextureMap Type="Diffuse" Name="boat" />
    </MTSMaterial>
  </MTSInstance>

  <MTSTimeElem Type="MTSImageStream" On="1"
  Path="images/boat.jpg" >
    <Target Name="MTSTexture.boat" />
  </MTSTimeElem>
</MTSScene>
```

© 2003 Viewpoint Corporation. All rights reserved.
Adding Flash to a Viewpoint Technology Web Application

2 Near the end of the file, find an MTSTimeElem element that loads the JPEG file you applied in Scene Builder. For example:

```xml
<MTSTimeElem Type="MTSImageStream" On="1" Path="images/boat.jpg">
  <Target Name="MTSTexture.boat" />
</MTSTimeElem>
```

3 Give the MTSTimeElem element a name and replace the Type and Path values as shown in the following example:

```xml
<MTSTimeElem Name="myflash" Type="SWFView" On="1" Path="flashfile.swf">
  <Target Name="MTSTexture.boat" />
</MTSTimeElem>
```

4 Following the MTSCamera tags, add an MTSTextureMap tag with X, Y, and HasAlpha values, like the following:

```xml
<MTSTextureMap Name="boat" x="128" y="128" HasAlpha="1" />
```

More Tips

After replacing the JPEG image with a Flash image, the JPEG image is still loading and taking up memory. Once the Flash image is in place, open the original JPEG image file and maximize the compression settings on that image. If the image is complex, you can replace it with a solid black or white color and save it. This method allows for the greatest compression.

If the replacement Flash image doesn't look sharp with x="128" y="128", you can increase the resolution by increasing the number to the next power of 2 (that is, x="512" y="512"), but at a slight memory cost. This controls the mapping coordinates on the object.

Now you can use Viewpoint Media Publisher to create a web page with your Viewpoint Technology-enabled scene embedded. For more information, see the Viewpoint Media Publisher User Guide available from Viewpoint Developer Central at http://developer.viewpoint.com/.

Preloading a Flash Movie

Using a Flash file as a preloader is a good way to engage the attention of a user and convey information while the main scene loads. Because the Flash and . mtx files are small, the preloader scene loads and displays very quickly.
While the Flash movie in the preloader scene is playing, the main .mtx file is loaded and calls .mts file, which contains scene contents set to collapse. The main scene files are larger and take longer to load. Once the main scene has fully loaded, the Flash movie stops and the main scene is uncollapsed.

**To create an .mtx file that preloads a Flash movie**

1. Using any XML or text editor, copy and paste the following code into your file:

   ```xml
   <?xml version="1.0"?>
   <MTSScene Version="4">
   <MTSTimeElem Type="SWFView" Name="preload" On="1" PostAnimator="1">
     <Target Name="MTSTexture.BackBuffer" />
   </MTSTimeElem>
   <MTSInteractor>
     <MTSHandle Event="AnimationDone:preload" Action="MTSLoadMTX" Path="my_scene.mtx" />
   </MTSInteractor>
   </MTSScene>
   ``

2. Replace "preload.swf" with the filename of your Flash movie.

3. Save the file as an .mtx file.

The MTSTimeElem loads and plays the Flash movie (preload.swf or the file you used to replace it). Loading a Flash movie percolates an AnimationDone event that you can reference in your .mtx file. The AnimationDone event occurs once the .swf file is loaded. Once you reference this event, you then use MTSLoadMTX to load the scene .mtx file, which will in turn load the .mts file.

The MTSInteractor element creates an interactor that will catch the AnimationDone event for the Flash movie (preload in this case) and load the .mtx file (final_audi.mtx in this case).

**Setting the Main Scene to Play Upon Loading**

You can use the same principles as described above for the second .mtx file. In this file, you will load the .mts file. Once it has been loaded, the .mtx file will catch the .mts file AnimationDone event and set triggers to play your main scene. Because the geometry loads progressively, you will want to collapse the entire scene while loading the .mts file.

**To edit your main .mtx file to load after the Flash preloader is done playing**

1. Open the main .mtx file for your scene using any XML or text editor:

2. Set the top-level instance (in this case final_audi) to Collapsed="1".

3. At the bottom of the file, add an interactor to catch the AnimationDone event that is percolated when the scene finishes loading, and add a set of Actions that will occur once you catch the AnimationDone. For example:

   ```xml
   <MTSAction Name="loadingdone">
   ```

   **Note:** You can replace the name "preload" with any name you want. Also, be sure to replace "my_scene.mtx" with the name of your .mtx file.
Once the second .mtx file is loaded, the two distinct .mtx files are now treated as one large .mtx file. You can make calls to stop animations in the first file from the second file. The interactor catches the AnimationDone event for the "final_audi" scene, and then calls the "loadingdone" action. The "loadingdone" action has several actions that will be carried out.

In this example, two of the actions are required, and two optional. There could be other optional actions. All the actions use SetProperty.

**Required Actions**

- The first action turns off the Flash movie, named "preload" in this case.

  ```xml
  <MTSAssignProperty Target="MTSTimeElem.preload::stop" Value="(mts_int)1"/>
  ```

  This action does a SetProperty of "stop" on this instance of preload. The value is an integer and it is set to 1 (trigger the stop immediately). "Stop" is used here rather than simply setting Loop="1" to make sure that the animation isn't interpreted as something playing in the background.

- The second action uncollapses the scene, making it visible. The action does a SetProperty of "clps" (collapse) on the instance of "final_audi". The value is an integer and is set to 0 (turn off collapse).

  ```xml
  <MTSAssignProperty Target="MTSInstance.final_audi::clps" Value="(mts_int)0"/>
  ```

**Optional Actions**

- The first optional action triggers a Flash movie that is going to play in the scene:

  ```xml
  <MTSAssignProperty Target="MTSTimeElem.flash2::trgr" Value="(mts_int)1"/>
  ```

  This action does a SetProperty of "trgr" (trigger) on the texture animation of "flash2". The value is an integer and is set to 1 (on).

- The second optional Action triggers an animation in the scene.

  ```xml
  <MTSAssignProperty Target="MTSTimeElem.start_ANIM::trgr" Value="(mts_int)1"/>
  ```

**Important:** The Action you call from AnimationDone must be above the Interactor in the .mtx file.
This action does a SetProperty of "trgr" (trigger) on the texture animation of "start_ANIM". The value is an integer and is set to 1 (on).

**Important:** Animations that are turned on while the file is preloading will not wait to be triggered, they will play before the geometry is expanded. Set all animations to On="0" and then trigger in your loadingDone action.
Chapter 7: Testing, Troubleshooting, and Deploying

About Testing: A Four-Phase Approach

Use these four steps to verify the correct appearance and functionality of your Viewpoint Technology content:

1. Upload your content to a test a web page.
2. Test all animations and interactivity in your Viewpoint Technology web application.
3. Test your content on different browsers and operating systems.
4. Troubleshoot any problems you encounter.

Uploading Your Content to a Test Web Page

Several 3D authoring applications let you publish your 3D model directly to a web page. If your application doesn’t include this export feature, you can use Viewpoint Media Publisher (available from Viewpoint Developer Central at http://developer.viewpoint.com/) to place your model into a web page.

To embed your 3D model into a web page using Media Publisher

1. Start Viewpoint Media Publisher.
2. From the HTML Template menu, choose generic.html.
3. Drag the .mtx file for your 3D model onto Media Publisher.
4  Type a name for the .html file, and then click Save.

5  Double-click the icon for the newly created .html file to view it in your default web browser.

   Viewpoint Media Publisher automatically creates buttons that you can use to test your Viewpoint Technology web application's animations, as described in the next section.

Testing Animations and Interactivity

If your animations are set to off (MTSTimeElem On="0") when creating your Viewpoint Technology-enabled web page using Media Publisher or your 3D application's export utility, the web page will automatically be created with buttons you can use to test animation.

**Note about On=:** If you set On="1" the animation runs when it is loaded. If you set On="0" the animation runs when it is triggered. In most examples, we use On="1", but for testing purposes, you may want to use On="0" so that you can manually trigger each animation.

To test your Viewpoint Technology web application's animations and
interactivity

1. Click each of the buttons created by Viewpoint Media Publisher or your Viewpoint Technology web application export utility.

2. Test standard navigation with mouse buttons:
   - Click and drag in the scene to rotate it (or each object, depending on how you authored it).
   - Right-click and drag the scene to resize it.
   - Press down both mouse buttons and drag the scene to move it.

3. Point to or click hot spots to ensure each behaves the way you intended.

Testing Viewpoint Technology Content Across Browsers and Operating Systems

To test your Viewpoint Technology web application for web readiness

1. Review your content on various browsers (Microsoft Internet Explorer 4.0 and 5.0; Netscape 4.08 or greater; AOL 4, 5, and 6).

   **Note:** If a MIME type is incorrect on the server, when viewing Viewpoint Technology content through a browser you may see text or images related to hot spots, but not see the 3D models, or you may see an error message "Unable to load XML file specified." For more information on MIME types, see the Deployment for System Administrators Guide.

2. Test your content on different operating systems (Windows® 98, ME, Windows NT® 4.x, Windows 2000; Macintosh® OS, and Classic browser on Mac OS X).

3. Look for the following in your Viewpoint Technology-enabled content:
   - Verify that your content loads.
   - Click all buttons to verify that there are no JavaScript errors. This verifies that animations are triggered correctly.
• Move models around to make sure they work. The illustration provides instructions on how to navigate Viewpoint Technology-enabled content.
Troubleshooting
This section lists some of the most common problems and suggestions for resolving them.

**MSTInterface.js (or mtsAxDetect.vbs) not found**
An error message tells you that either MTS3Interface.js or mtsAxDetect.vbs was not found. (In Internet Explorer, this message could look like a debug error.)

Possible solutions:

- Ensure that the files are in their proper locations.
- Check for capitalization inconsistencies. Make sure that the names of the files themselves and all references to these files use capitalization consistently.

**Image links are broken**
Possible solutions:

- Ensure that the files are in their proper locations.
- Check for capitalization inconsistencies. This is pertinent to Unix environments, but not Windows NT. The latter is not case-sensitive. Make sure that the names of the files themselves and all references to these files use capitalization consistently. In general, it's best to name referenced files you create in all lowercase.

"www.VIEWPOINT.com" watermark appears in front of Viewpoint Technology content
The web address "www.VIEWPOINT.com" appears in transparent gray letters in front of your Viewpoint Technology content.

Possible solutions:

- Be sure that the HTML is pointing to correct location for the Broadcast Key file.
- Check the Broadcast Key file (bkey.txt, the licensing key that allows Viewpoint content to display properly) to ensure the following:
  - Be sure you have the bkey.txt file, and that it's in the root directory for your content.
  - Be sure the Broadcast Key numbers contained in the file are valid.
  - Broadcast Keys must be updated for ZoomView, Flash integration, Poser, or any other special features.

**Message: "Unable to load XML file specified."**
This error message displays when visiting Viewpoint Technology-enabled content.

Possible solutions:

- Ensure that the files are in their proper locations.
- Check the path to see that it lists the XML file location accurately.
- Check for capitalization inconsistencies. Make sure that the names of the files themselves and all references to these files use capitalization consistently. In general, it's best to name referenced files you create in all lowercase.
Deploying Your Viewpoint Technology Web Application

To properly to place and enable your Viewpoint Technology web application on a server, make sure you have completed the steps in the following checklist. For more information about completing these steps, see the Testing and Deploying Viewpoint Technology Content Guide which is available on Viewpoint Developer Central at http://developer.viewpoint.com/.

Deployment Checklist

• Ensure all required Viewpoint Technology files, including resource files, are in the correct folders and are ready to deploy.

• Provide a first-time installation option from your Viewpoint Technology web application. Include automatic installation, for Viewpoint Media Player.

• Set MIME types for Viewpoint Technology files. For more information, see the Deployment for System Administrators Guide.

• License your Viewpoint Technology content with a Broadcast Key.

  Note: To publish Viewpoint Technology content on a website, the site publisher must complete a Broadcast License Agreement and acquire the associated Broadcast License Key (a "key"). Broadcast licenses are FREE for non-commercial use. For more information about obtaining and using Broadcast Keys with your Viewpoint Technology content, see the How To Buy page on the Viewpoint Developer Central.

• As applicable, place Viewpoint affiliate logos in your Viewpoint Technology web application.
Appendix A: Help, Resources, and Feedback

Viewpoint Developer Central: A Complete Resource

Viewpoint Developer Central is a complete resource for Viewpoint Technology content developers. At this website, you can access Viewpoint applications, user guides, downloadable example files, support, production tips, and techniques – to name just a few of the offerings there.

Access to Viewpoint Developer Central to:

- **Get Assistance** For questions about using Viewpoint Technology, click Forums under Support in the left navigation bar.
- **Get Examples** Click Examples & Tips in the left navigation bar.
- **Subscribe to the Viewpoint Developer Newsletter** Learn new production tips and techniques for creating 3D and rich media content for the web with Viewpoint Technology. Click Newsletter in the left navigation bar.
- **Give Feedback About Viewpoint Applications** Viewpoint Corporation values your feedback. Direct your comments and suggestions to the Viewpoint Forums.

You can also visit the Viewpoint Corporation website for company news, links to websites featuring Viewpoint Technology, and more.

Download Viewpoint Applications, Guides, and Examples

Viewpoint Developer Central is updated continuously with the latest versions of applications, user guides, and examples. Find links to the following in the left navigation bar.

**Viewpoint Applications**

You can download Viewpoint applications free of charge. The following applications are available for download:

- **Viewpoint Media Player** — The web browser plug-in necessary to view Viewpoint content with Netscape Navigator or Internet Explorer.
- **Viewpoint Scene Builder** — An essential application for assembling a scene and publishing it in .html/.mtx/.mts format.
- **Viewpoint Media Publisher** — An application enabling you to quickly create Viewpoint Technology web applications from Viewpoint media files (.mtx/.mtz) by embedding them in web (html) pages or running transformations on .mtx (xml) files through built-in XSLT support.
- **Viewpoint Stream Tuning Studio** — An application for reducing .mts file sizes, enabling optimized 3D scenes rendered on a web page to stream quickly and retain visual integrity.
- **Viewpoint Control Panel** — A utility for checking, installing, and removing individual Viewpoint Media Player components.
## User Guides

For more information on Viewpoint Technology and related utilities, check out these documents on [Viewpoint Developer Central](#):  

<table>
<thead>
<tr>
<th>See this document</th>
<th>To learn about</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viewpoint Technology Feature Overview</td>
<td>What Viewpoint Technology is and how you can use rich media components-including 2D photos and drawings, 3D models, animation, ZoomView images, Macromedia Flash movies, video, text, and audio-to create rich media content.</td>
</tr>
<tr>
<td>Using 3D in Viewpoint Technology</td>
<td>Describes how to create a basic 3D Viewpoint Technology web application with simple interactivity and Flash effects. You'll learn how to take content created in a 3D modeling application, enhance it with Viewpoint Technology, and publish it to a web page.</td>
</tr>
<tr>
<td>Using High-Resolution 2D Graphics in Viewpoint Technology</td>
<td>This guide is a comprehensive reference for ZoomView, a technology that permits the deployment of large, print quality images online. The guide explains the three authoring paths to create ZoomView images, which include Adobe® Photoshop® 7, ZoomView Builder, and ZoomView Hosted Solutions. It also describes how to add advanced features to your ZoomView scenes by making changes to XML code in your .mtx file.</td>
</tr>
<tr>
<td>Viewpoint Stream Tuning Studio Guide</td>
<td>Explains how to use Viewpoint Stream Tuning Studio to optimize uncompressed .mts files from Viewpoint Scene Builder and other applications that export .mts files for Viewpoint Technology scenes.</td>
</tr>
<tr>
<td>Viewpoint Media Publisher Guide</td>
<td>New features of Viewpoint Media Player provide a fast and convenient path to the web for Viewpoint Technology content. Viewpoint Media Player enables you to quickly create Viewpoint Technology web applications from Viewpoint Media Files (.mtx/.mtz) by embedding them in web (html) pages or running transformations on .mtx (xml) files. This guide includes step-by-step instructions for creating Viewpoint Technology-enabled web pages, customizing html templates, customizing .mtx files with XSLT, and publishing Viewpoint content.</td>
</tr>
<tr>
<td>Using Vector Graphics in Viewpoint Technology</td>
<td>Describes how to integrate Macromedia® Flash™ and Scalable Vector Graphics (SVG) with other rich media, such as animated 3D objects, high-resolution photographs, object movies, and panoramas into Viewpoint Technology scenes.</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Viewpoint Technology: Technical Overview</strong></td>
<td>Describes the architecture of a Viewpoint Technology scene, including the key components that comprise it: objects, animations, events, and properties. You'll learn how to combine these components and organize them in the Viewpoint Technology scene .mtx description file.</td>
</tr>
<tr>
<td><strong>Deployment for System Administrators</strong></td>
<td>Steps that Administrators can take to set up a web server with Viewpoint Technology content.</td>
</tr>
<tr>
<td><strong>Testing and Deploying Viewpoint Technology Content</strong></td>
<td>Describes how to test and prepare your Viewpoint Technology content for deployment. Details the MTS3Interface, the auto-installer, and their usage, making it an invaluable resource for Viewpoint content producers.</td>
</tr>
<tr>
<td><strong>Viewpoint XML Reference Guide</strong></td>
<td>A powerful reference tool containing the most up-to-date information and examples for the XML tags and properties that control a Viewpoint Technology scene. Viewpoint Technology uses XML scripted in an .mtx file to orchestrate the elements of a scene, including animation, interactivity, and loading of files, as well as ZoomView and HyperView functionality.</td>
</tr>
</tbody>
</table>
# Glossary

**3D (three-dimensional)**

An object or volume that exists in the dimensions of width, height, and depth.

**action**

The transition code (the action) is executed if the start state of the transition matches the current state maintained by the interactor and the current event is equal to the transition event.

**animation**

A motion or transition added to a media atom or a group of media atoms over time. Examples include an object moving around a scene, transitions from one color or texture to another, or an object becoming visible.

**broadcast key**

A unique alphanumeric string issued by Viewpoint Corporation to companies or individuals licensed to broadcast Viewpoint Technology content. The string is stored in a text (.txt) file that is referenced by Viewpoint Technology-enabled web pages. Viewpoint Technology content without a broadcast key displays with a watermark.

**bump mapping**

A method of displaying textures not as a smooth surface, but as a rough surface that responds to different angles of illumination.

**camera**

The view from which a scene is rendered.

**element**

The complete statement of an XML command contained between an opening and closing tag. Elements include attributes and values and may contain nested elements, also known as subelements.

**geometry**

An entity that defines all polygons making up an object. (Also called a mesh or simply an object.)

**global**

Describes properties added to an entire scene.

**hot spot**

An area in a scene made up of a procedural shape (usually invisible) and created in Scene Builder. Hot spots are generally used to define a 3D area that when interacted with displays a text annotation, texture, or Flash movie. For instance, when a user points to a hot spot, a text-based annotation may appear.

**interactors**

Elements that allow the user to alter or interact with the scene by clicking or rolling over certain areas within the scene. The programmer defines the interactors in the XML code. OnClick and OnEn are examples of interactor events.

**light map**

An image that determines how light interacts with and scatters on the surface of an object. Material properties such as diffusion, specularity, and reflection are captured in the lightmap. The lightmap in any Viewpoint scene is what the
camera sees in any reflective materials of an object. Any spherical image can be used as a lightmap image.

map
To apply a 2D image onto the surface of an object.

materials
Surfaces added to the mesh to give it a finished appearance unlike wireframe rendering.

media atoms
Components of a Viewpoint scene: 3D objects, material properties, sound, object movies, animators, interactors, and the definition of the 3D environment (that is, panoramas or the maps of environmental lightmaps).

mesh
See geometry.

MTS
A binary resource file containing all geometry, materials, and texture information for a Viewpoint Technology scene. MTS is an open-specification 3D file format developed by MetaCreations and Intel Architecture Labs. A key Viewpoint Technology feature is scalability, which automatically maximizes performance and resolution of content according to the available processing power of the user’s PC. As a result, web designers can set the detail (frame rate) of these 3D objects to ensure consistent, high-quality viewing. Viewpoint Technology also progressively streams content, allowing users to experience 3D objects from the moment downloading commences, similar to streamed audio or video files. A published Viewpoint Technology scene consists of an .mts file and an .mtx file.

MTX
A Viewpoint XML scene file that contains the hierarchical relationships between objects and other elements in the scene. This file is the script for staging the scene elements and usually references an .mts file.

MTZ
A compressed form of an .mtx file and the preferred format for web-enabled Viewpoint content. Complex animations in an .mtx file can make file size large. Compressing these large .mtx files enables fast downloading of Viewpoint scenes.

properties
Attributes of a media atom.

render
The transformation of 3D data into 2D frames for display on a computer screen.

rotation
Moving an object around a specific center and axis.

scene
The highest level of the Viewpoint Technology hierarchy (MTSScene tag in XML). Scene contains all elements of the .mtx and .mts files.

SWF
The Macromedia Flash™ movie file extension. May be pronounced “swif.”

texture
A picture on the surface, usually a JPEG or similar image file. This image file is rendered over polygons to give the object a realistic-looking surface.

tiling
The method of repeating a texture more than once across an object or part of an object. A tiled texture looks best if its edges
seamlessly match up with each other, top to bottom and side to side. Tiling is a common method of using the smallest texture possible to cover a large area, such as a texture of a brick tiling across a large polygon or object to create an entire brick wall.

**transform**  
Transforms are position, rotation, and scale.

**translate**  
To move the object along the x, y, or z axis in the scene.

**Viewpoint Technology**  
Viewpoint Corporation’s unique technology that streams 3D and rich media content (media atoms) over the Internet via Viewpoint Media Player, a web browser plug-in.

**Viewpoint Media Player**  
The web browser plug-in necessary to view Viewpoint Technology content with Netscape Navigator or Internet Explorer.

**widget**  
An area in a scene made up of a procedural shape (usually invisible) and created in Scene Builder. Widgets are generally used to define a hot spot that when interacted with displays a text annotation, texture, or Flash™ movie. For instance, when a widget is rolled over, a text annotation may appear.

**XML**  
Extensible Markup Language. A markup language for documents containing structured information with instructions for content (words, pictures, etc.) and the role that content plays (for example, content in a section heading has a different meaning from content in a footnote, figure caption, or database table). Viewpoint Technology uses XML to define all properties of a scene.