Viewpoint

Viewpoint
Media Publisher
User Guide

Version 3.0.7.x

May 16, 2001

Viewpoint Media Publisher User Guide

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

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.

Authors: Andrew Cook, Lars Olson, and Carolyn Gronlund

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

Contents

Overview

A Major Upgrade From Viewpoint MTX 2 HTML

Viewpoint Media Publisher is a major upgrade from Viewpoint MTX 2 HTML that is more adaptable to the needs of Web content creators. As the name Media Publisher implies, this new tool provides a fast and convenient path to the Web for Viewpoint Experience Technology (VET) content. Media Publisher enables you to quickly create Web (.html) pages from Viewpoint Media Files (.mtx/.mtz).

New Features Make Media Publisher a Powerful Production Tool

Media Publisher's new features give you the power to easily create attractive Web pages enabled for Macintosh and Microsoft Windows users. This upgrade includes:

- **Easily customizable HTML templates** that can be used as is or with your own design elements, such as background images, stylesheets (.css files), logos, navigation, and text.
- **Automatic cross-platform and Web browser compatibility** with an embedded JavaScript file, MTS3Interface.js, Web pages you create in Media Publisher behave consistently under these systems: Microsoft Windows 95 or greater with Netscape Navigator 4.07 or greater (not version 6.0), Microsoft Internet Explorer, or AOL 4.0, 5.0, or 6.0; or Macintosh OS 8 with Netscape Navigator 4.07 or greater (not version 6.0) or AOL 4.0, 5.0, or 6.0.
- **Automatically generated resource folders** where you can keep track of the resource files for each Web page, including logos and background images, and VET scene.
- **Viewpoint Media Player (VMP) minimum version** you can set to ensure users are updated to the version of VMP that supports the features in your scene.
- The capability to embed **multiple scenes in a single Web page** (for power users). To find out about how to embed multiple scenes in a single Web page, refer to *About Viewpoint MTS3Interface.js* in the Media Player "docs" folder. Double-click this file to open and read it in your default Web browser.
- **Periodic releases of new HTML templates** to give content creators instant access to new Web page designs.

In addition, Media Player retains MTX 2 HTML's useful features:

- Automatic creation of buttons to trigger animations in an .mtx/.mtz file.
- Customizable scene window height and width settings.
- The wireframe.html template that is especially useful for game designers.

System and Software Requirements

Viewpoint Media Publisher Minimum System Requirements

- Windows 95 or NT
- 200 MHz Pentium processor
- Netscape Navigator 4.07, Microsoft Internet Explorer 4.x, or AOL 4.0, 5.0, or 6.0

> Note: The current version of Viewpoint Media Player does not support Netscape Navigator 6.0.

- 32 MB system RAM
- 10 MB free hard disk space
- Color display (24-bit recommended)
- 1024 × 768 monitor resolution (recommended)

Viewpoint Media Player Minimum System Requirements

Viewpoint Media Player (VMP) is designed to stream Viewpoint Experience Technology under the following *minimum* system configurations:

Windows

- Windows 95 or NT
- Pentium 166 (Pentium II recommended)
- Netscape Navigator 4.07, Microsoft Internet Explorer 4.x, or AOL 4.0, 5.0, or 6.0

> Note: The current version of Viewpoint Media Player does not support Netscape Navigator 6.0.

- 256 color display (24-bit recommended)
- 28.8 Kbps modem
- 5 MB free disk space
- 32MB RAM (64MB recommended)

Macintosh

- Mac OS 8.5 (not OS X)
- PowerPC 604 processor
- Netscape Navigator 4.7 (not 6.0) or AOL 4.0
- 256 color display (millions of colors recommended)
- 56.6 Kbps modem
- 5 MB free disk space
- 128 MB RAM

Required Software

You can download the latest versions of Viewpoint Media Publisher, Viewpoint Media Player, and other Viewpoint tools free of charge from the Viewpoint Developer Central Web site.

- Viewpoint Media Publisher
- Viewpoint Media Player (VMP) The Web browser plug-in that streams rich media and is required for viewing and interacting with VET.
- Netscape Navigator 4.07 or later, Microsoft Internet Explorer 4.x or later, or AOL 4.0 or later.

> Note: The current version of Viewpoint Media Player does not support Netscape Navigator 6.0.

Help, Resources, and Feedback

Viewpoint Corporation's newly updated Developer Central Web site is the complete resource for content creators using Viewpoint Experience Technology. At Viewpoint Developer Central, you can

- Go to the **Forum** for answers from Viewpoint staff and other VET developers. You can also give Viewpoint **feedback** here and take part in the development of this exciting, new technology.
- Subscribe to the **Viewpoint Content Creator Newsletter**, a bi-weekly publication containing product information, tips and tricks to make authoring easier, information on new Viewpoint Media Player releases, and more. You can also read back-issues of the newsletter.
- Download **Viewpoint applications** (including **Viewpoint Media Player**), **user guides**, and **tutorials** free of charge.
- Download **demos**, regular **media atom freebies** (such as a texture map), and **affiliate logos** to identify your Viewpoint content.
- Read all about VET: feature descriptions, **FAQs**, weekly content creation tips, white papers, and more.

How VET-Enabled Web Pages Work

About Broadcast Keys

Broadcast License Keys are Viewpoint's equivalent of a license or serial number required to display Viewpoint Experience Technology (VET) content on a Web site. If your .html file does not reference a Broadcast License Key, your VET content displays, but with a watermark (the name VIEWPOINT).

Obtain a Broadcast Key (a unique alphanumeric string that you receive when you are licensed to broadcast VET content) by first completing the Broadcast License Request form on Viewpoint's Web site. Both Flash and ZoomView content require a special key. Be sure to check the appropriate options when applying for your key.

Options for Storing and Referencing Broadcast Keys

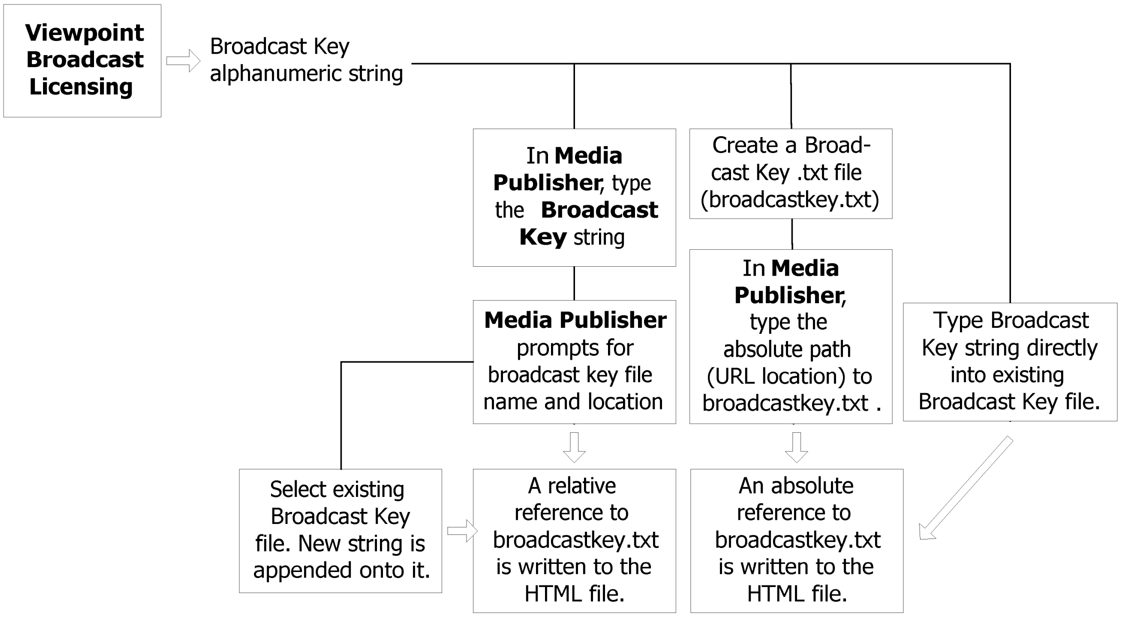
A Broadcast Key must be stored in a .txt file referenced by the Web page (.html file) containing VET content. You have two primary options for storing the Broadcast Key .txt file:

- Relative reference from the .html file. This option requires that you keep track of the location of the .txt file in relation to the .html file that references it.
- Absolute reference from the .html file. Use one Broadcast Key file, and reference it from multiple .html files. A URL address (for example, http://www.CorporateURL.com/) is the best choice for a live website.

> Note: You can store Broadcast Keys for multiple URLs in the same BroadcastKey.txt file.

The following diagram shows options for creating and storing a BroadcastKey.txt file. To find out more about Broadcast Keys, go to the Viewpoint Developer Central Web site.

Options for Storing a Broadcast Key


Viewpoint Broadcast Licensing
Broadcast Key alphanumeric string
In Media Publisher, type the Broadcast Key string
Create a Broadcast Key .txt file (broadcastkey.txt)
Media Publisher prompts for broadcast key file name and location
In Media Publisher, type the absolute path (URL location) to broadcastkey.txt .
Type Broadcast Key string directly into existing Broadcast Key file.
Select existing Broadcast Key file. New string is appended onto it.
A relative reference to broadcastkey.txt is written to the HTML file.
An absolute reference to broadcastkey.txt is written to the HTML file.

Enter the Broadcast Key string directly into Media Publisher to generate a Broadcast Key file. Or, create your own file or use an existing Broadcast Key file, and then specify an absolute reference to its location.

MTS3Interface.js Enables VET Across Platforms and Allows Multiple Scenes in a Single Web Page

MTS3Interface.js is a JavaScript resource file referenced by Web pages you create in the current release of Media Publisher. With MTS3Interface.js, your VET-enabled Web pages display consistently whether viewed in Microsoft Internet Explorer or Netscape Navigator, and on the Macintosh OS or Microsoft Windows operating system. In general, MTS3Interface.js does not require any special coding for you to take advantage of its features.

Power users can read more about MTS3Interface.js features and capabilities by double-clicking *About Viewpoint MTS3Interface.js* in the Media Player Docs folder. You can open and read this guide in your default Web browser.

Note: The MTS3Interface.js file copied in the resources folders is an optimized version. If you want to read a verbose version of the file, open the MTS3Interface.js file in the About Viewpoint MTS3Interface.js folder in the Media Player Docs folder.

VET Web Pages Adapt to Your Users' OS and Browser

The MTS3Interface.js file located in resources folder of each Media Publisher created Web page provides a uniform standard for Viewpoint JavaScript calls. This ensures that VET scenes display properly with Microsoft Internet Explorer and Netscape in either a Windows or Mac OS environment. (Viewpoint is the first company to integrate scripting between MacOS IE and MacOS IE plugins.) It's no longer necessary to have customized JavaScript code for specific browsers and operating systems.

MTS3Interface.js also handles all calls and messages returned by the Viewpoint Media Player (VMP) Web browser plug-in.

Important: MTS3Interface.js contains operators that control how your VET-enabled Web page displays. Do not edit this file.

Automatic Installation of Viewpoint Media Player Via the Netscape Browser

In previous versions of MTX 2 HTML, the trigger.js file triggered the automatic download of the Viewpoint Media Player (VMP) from Netscape browsers. Now, MTS3Interface.js includes this code from trigger.js.

Note: MTS3Interface.js triggers a *first* download of the Viewpoint Media Player via a Netscape browser. VMP updates automatically after that.

Multiple VET Scenes Allowed on a Web Page

Power users can create Web pages with mutliple VET scenes embedded simultaneously. To find more, refer to *About Viewpoint MTS3Interface.js* in the Media Player "docs" folder. Double-click this file to open and read it in your default Web browser.

Automatic Storage of Web Page Resources

When Media Publisher creates an HTML page, it also automatically creates a resource folder in the same location containing any files from the resources folder of the HTML template you used. The new folder has the same name as the HTML file and ends with *_resources*.

Tip: On your computer's hard drive, create a project folder to contain your Viewpoint Media Files, Web pages you create, and their resource folders.

Creating VET-Enabled Web Pages

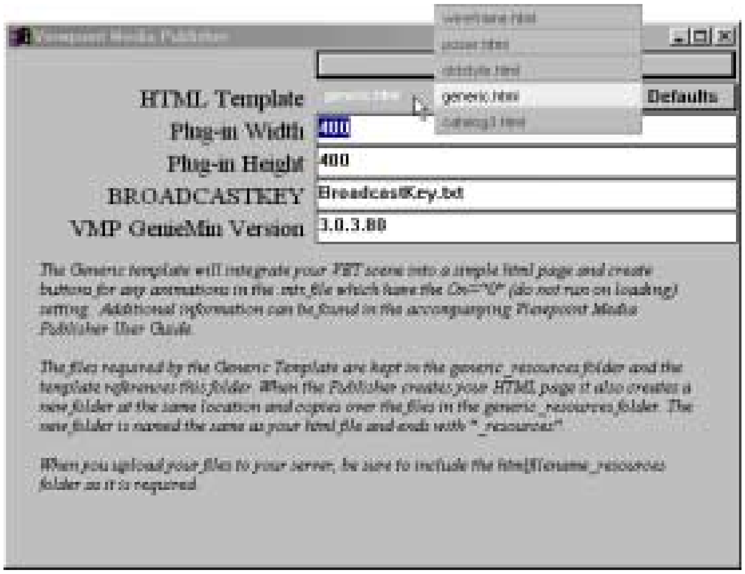
Basic Steps for Creating a VET-Enabled Web Page

Embedding VET scenes into Web pages has never been easier than with Viewpoint Media Publisher. Follow these steps:

1 From your 3D modeling application or Viewpoint Scene Builder, save or publish your scene as **Viewpoint Media Files** (.mts and .mtx/.mtz format).

2 Launch Viewpoint Media Publisher.

3 From the **HTML Template Menu**, choose the template you want to use.



HTML Template: generic.html
Plug-in Width: 400
Plug-in Height: 400
BROADCASTKEY: BroadcastKey.txt
VMP GenieMin Version: 3.0.3.80

The Generic template will integrate your VET scene into a simple html page and create buttons for any animations in the .mtx file which have the On="0" (do not run on loading) setting. Additional information can be found in the accompanying Viewpoint Media Publisher User Guide.

The files required by the Generic Template are kept in the generic_resources folder and the template references this folder. When the Publisher creates your HTML page it also creates a new folder at the same location and copies over the files in the generic_resources folder. The new folder is named the same as your html file and ends with "_resources".

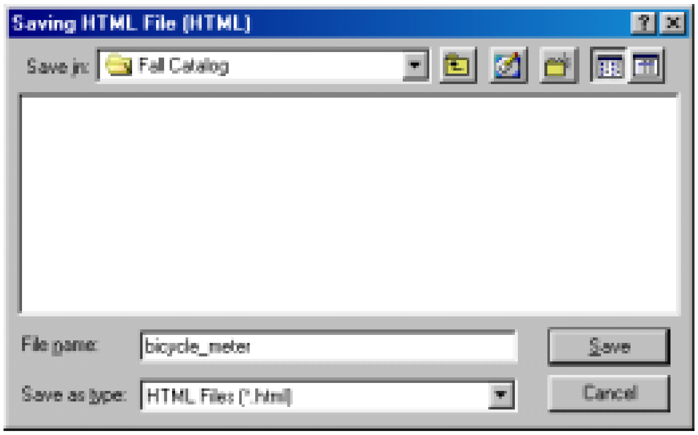
When you upload your files to your server, be sure to include the htmlfilename_resources folder as it is required.

Important: If your content uses ZoomView or Flash Compatibility*, you must use the zoomview.html or zoomviewoptimized.html templates. If your content uses Poser, you should use the Poser template or one of the ZoomView templates.

4 Drag a scene’s .mtx file onto Media Publisher.

Power User Tip: To find out about how to embed multiple scenes in a single Web page, refer to *About Viewpoint MTS3Interface.js* in the Media Player Docs folder. Double-click this file to open and read it in your default Web browser.

5 Type a name for the .html file, and then click **Save**.



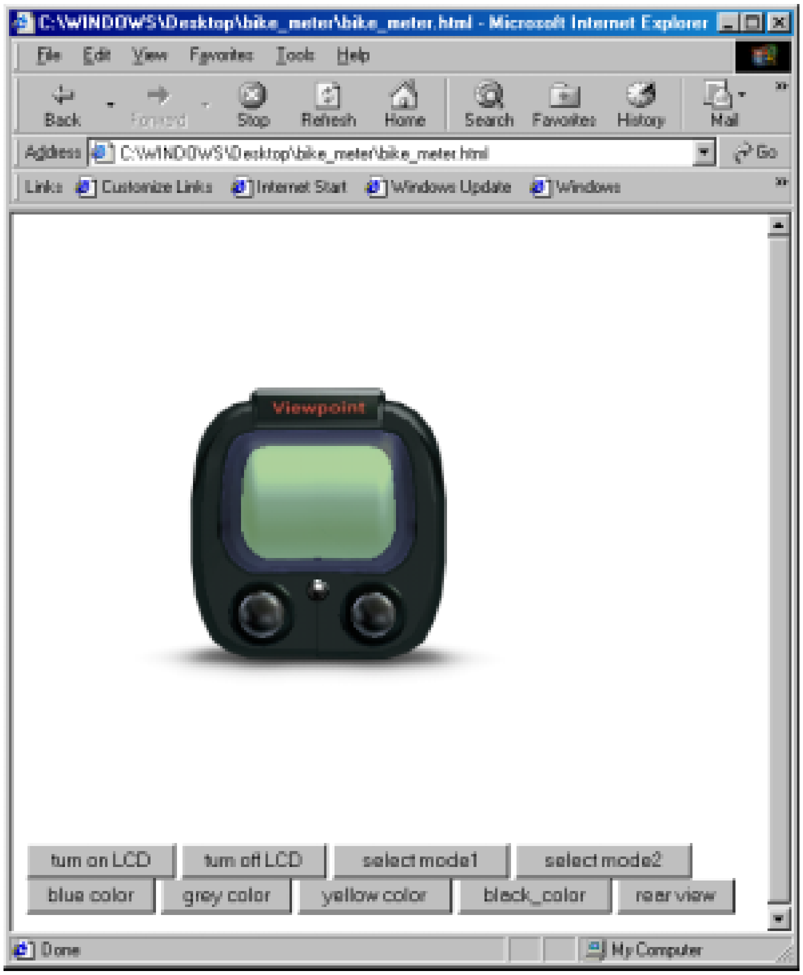
Saving HTML File (HTML)

Save in: Fall Catalog

File name: bicycle_meter

Save as type: HTML Files (*.html)

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



C:\WINDOWS\Desktop\bike_meter\bike_meter.html - Microsoft Internet Explorer

Address: C:\WINDOWS\Desktop\bike_meter\bike_meter.html

Repeat steps 4 and 5 to create additional Web pages.

7 **Register MIME types on servers.** This is a very important step to carry out before deploying content on your web server. Viewpoint recommends that you register .mtx, .mtz, .mts, and .mzv as MIME types. For more information, see Appendix: Registering MIME Types.

Working With Animation Buttons

Media Publisher creates a button for each animation for which On="0" (off). You can remove these buttons completely or use custom icons for them.

Use Custom Button Icons

In the following code example, *a* is a JavaScript variable, *animation1* refers to the unique name for the animator, and *button.jpg* refers to an image file for the button icon:

```
<p><a href="#" onclick="a.TriggerAnim('animation1')"><img src="button.jpg"
width="100" height="50" border="0"></a>
</p>
```

For each button you want to customize, add this code to the body of the .html file generated by Media Publisher. Be sure to include the unique animator name and image file name for the button.

Remove Media Publisher-Generated Buttons

You can remove some or all of the animation buttons generated by Media Publisher. This is useful if your animations are already triggered by OnClicks on widgets in the scene, for instance, and you don't want or need buttons to trigger the animations.

1 Open the Media Publisher generated .html file in a text editor.

2 Search for the `<form>` element.

Following is the <form> element for the bicycle meter pictured later in this guide:

```
<form>
<input name="turn_on_LCD" type=button  value="turn_on_LCD"
   onclick="triggeranimation('turn_on_LCD')">
<input name="turn_off_LCD"   type=button  value="turn_off_LCD"
   onclick="triggeranimation('turn_off_LCD')">
<input name="select_mode1"   type=button  value="select_mode1"
   onclick="triggeranimation('select_mode1')">
<input name="select_mode2"   type=button  value="select_mode2"
   onclick="triggeranimation('select_mode2')">
<input name="blue_color"  type=button  value="blue_color"
   onclick="triggeranimation('blue_color')">
<input name="grey_color"  type=button  value="grey_color"
   onclick="triggeranimation('grey_color')">
<input name="yellow_color"   type=button  value="yellow_color"
   onclick="triggeranimation('yellow_color')">
<input name="black_color" type=button  value="black_color"
   onclick="triggeranimation('black_color')">
<input name="rear_view"   type=button  value="rear_view"
   onclick="triggeranimation('rear_view')">
</form>
```

3 Look for the button(s) you want to delete. You may delete the entire `<form>` element, if you don't want to use any of the buttons generated by Media Publisher.

4 Save the edited .html file, and then double-click its icon to open it in your default Web browser.

Media Publisher Templates

About the Templates

Media Publisher installs with these templates located in the HTML Support folder:

- generic.html This is the standard template. New templates you create should be based on this template.
- catalog3.html Designed for an online catalog, this template creates a Web page showing one product and includes placeholders for a logo, text describing the product, a link to Help for Viewpoint Media Player, as well as navigation placeholders.
- wireframe.html This template includes JavaScript to change the render mode between default and wire (wireframe). The template also creates a button to trigger the render mode change.
- poser.html This template includes *ComponentMinimumVersion="50332936"* in both the embed/object tag. This forces the Viewpoint Media Player to update the Scene Component to the 3.0.5 version and is required if your content is created using Poser from Curious Labs.
- zoomview.html This template includes "GenieMinimumVersion="50333440", "ComponentMinimumVersion="50333440", HostMinimumVersion="50333440". This forces updating of the Component Manager, SceneComponent and Host (AXMetastream.dll) to version 3.0.7. ZoomView, Flash, and procedural cursors require the Genie and Component minimum versions. Additionally, the procedural cursors also require the Host minimum version to be 3.0.7.
- zoomviewoptimized.html This template includes everything in the zoomview.html template and, in addition, supports the use of absolute references to any required files. This helps optimize server performance by avoiding the duplication of resource files common to all VET scenes.
- oldstyle.html This template places the object/embed tag and the Viewpoint JavaScript functions directly into the HTML page.

> Warning: The oldstyle.html template is provided for backward compatibility reasons and is not recommended for new projects.

“Skins” Define the Look of Your Web Pages

In the case of Viewpoint Media Publisher, a skin is the overall look of a Web page as determined by design elements, such as a stylesheet (.css file), background color, background image, and logo image. Each Media Publisher template has a different skin that you can easily adapt to meet your design needs.

For more information, see “Working With Template Skins.”

Free Templates From Viewpoint

Periodically, Viewpoint will release new Media Publisher templates for download from the Viewpoint Developer Central Web site. Use these templates as they are, or adapt them to your content needs.

Template Masters are Easy to Customize

Each custom template design from Viewpoint (catalog3.html is the one that installs with Media Publisher) comes with a master version. The master does not contain any of the template code necessary for Media Publisher, but is easier to edit in a Web authoring application. In the HTML Support folder, the master template is in a folder named the template name with *_master* (underline plus the word *master*) appended.

The code that makes an .html file a Media Publisher template can be added later as described in “Use a Web Authoring Application to Create a Custom Template.”

Customizing Templates

By customizing templates, you adapt Viewpoint Media Publisher to your production needs. The easiest way to change a Media Publisher template skin is to modify an existing template using your custom art and stylesheet (.css file). You can also use an .html file you create as a template by inserting code from an existing Media Publisher template.

Tip: Use standard naming conventions for Web page design elements, such as background images and logos.

- Tiled background image: Add _*x* or _*y* (underline plus *x* or *y*) to the filename to indicate whether the image is tiled along the x or y axis.
- Other images: At the end of the filename, append the image size (in pixels). For instance, image600x800.jpg.

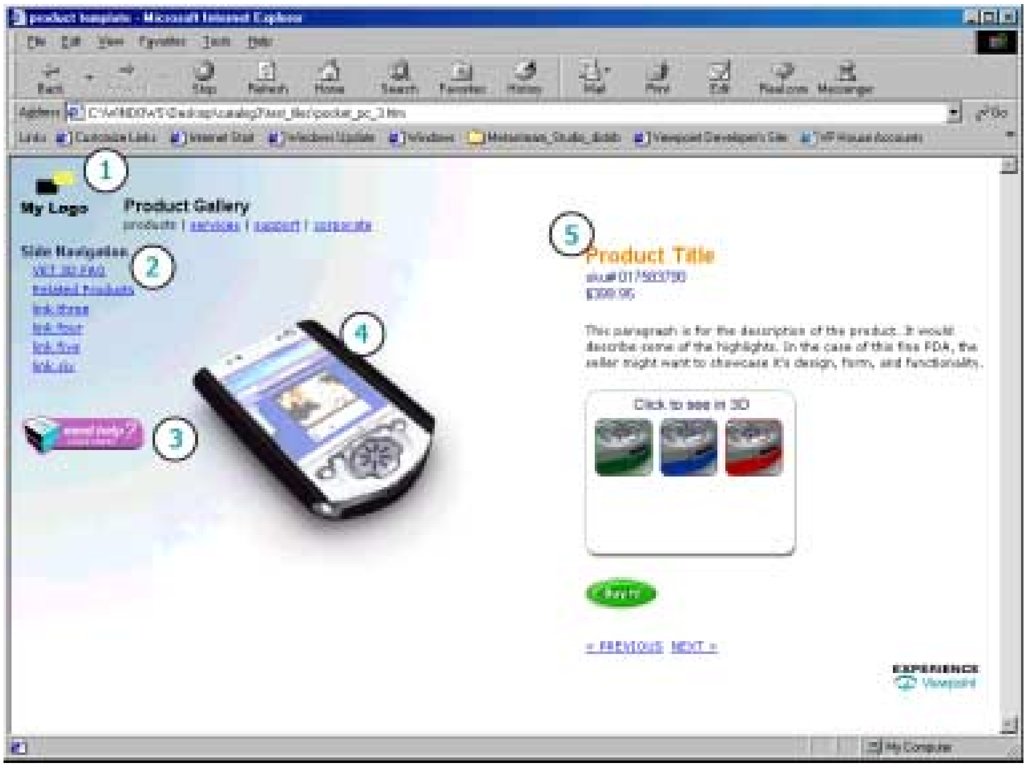
Working With Template Skins

Easily adapt the look of an existing template by substituting the files of a new skin with little or no manipulation of the HTML code. For example, the catalog3.html template file can be changed in the following ways:

- Style sheets You can change fonts, colors, and background graphics through the use of a cascading stylesheet (.css file), the easiest way to quickly change the appearance of a Web page.
- Graphics Any graphic in the template can be replaced, including logos and background images. Keep in mind that a tiling background image requires transparent .gif images to show without a solid background.
- Removal and addition of sections/subsections The template is sectioned into 4 tables. The Viewpoint Experience table at the bottom of the Web page must remain, but other tables and information are optional.

Parts of a Media Publisher Template Skin

The following illustration shows the parts of the catalog3.html template:



sku# 017583790
$399.95

This paragraph is for the description of the product. It would describe some of the highlights. In the case of this fine PDA, the seller might want to showcase it's design, form, and functionality.

1 Company header This upper part of the page is for company navigation and a logo. It's a separate table for easy cut and paste of your existing upper navigation.

2 Side navigation This serves as sub-navigation for section 1. This can be used to link to FAQs, additional product information, related products, and so forth.

3 Help This button spawns a layer with instructions for manipulating the Viewpoint 3D object. Clicking on the button again makes the layer disappear. This can be customized by creating your own .gif or .jpg.

4 VET content This part of the Web page is where the VET object is embedded.

5 Product information The title, sku or reference number, price, description, interaction palette, purchase button, and next/previous navigation. The interaction palette can allow the user to change colors on the 3D object, animate some section of the object, or move the object to preset views.

Tip: Each custom template design (catalog3.html is the one that installs with Media Publisher) from Viewpoint comes with a master version. The master does not contain any of the template code necessary for Media Publisher, but is easier to edit in a Web authoring application. In HTML Support, the master template is in a folder named the template name with *_master* (underline plus the word *master*) appended.

Change a Skin

1 First, create a Web page using the original template to see what design elements you want to change.

2 Create a project folder, and then copy into it the scene files you want to publish to the Web.

3 Make a backup copy of the resources folder for the HTML template you want to customize.

4 Create the image files and stylesheet you want to use, give them the same names as the corresponding elements in the original template, and then copy them over the original skin files in the resources folder for the template you are customizing.

If you give your custom design elements file names that are different from the design elements specified in the original template, you must enter the new names in the appropriate fields each time you create a Web page in Media Publisher. However, in this case, if you are creating multiple Web pages with the same custom skin, it's best to open the template file in a text editor and change the references to the design elements to match the new file names.

5 On your computer's desktop, double-click **My Computer**. Locate the project folder you created in step 2.

6 Launch **Viewpoint Media Publisher**.

7 From the **HTML Template Menu**, choose the template you want to use.

If you named your design elements to match the original template or if you edited the template to include your custom names, you don't need to change the fields containing the design element names.

Otherwise, type the exact file names for your custom design elements in the appropriate fields.

8 Drag a scene's .mtx file onto **Media Publisher**.

> Power User Tip: To find out about how to embed multiple scenes in a single Web page, refer to *About Viewpoint MTS3Interface.js* in the Media Player Docs folder. Double-click this file to open and read it in your default Web browser.

9 Type a name for the .html file, and then click **Save**.

In the project folder, double-click the newly created .html file to view it in your default Web browser.

Repeat steps 8 and 9 to create additional Web pages.

Use a Template Master as a Design Shortcut

Each custom template design from Viewpoint (catalog3.html is the one that installs with Media Publisher) comes with a master version that you can easily open and edit in a Web authoring application. The master does not contain any of the template code necessary for Media Publisher.

The benefit of using the master, as you will quickly see if you try to open a template file in a Web authoring application, is that you can work with the design freely without the Viewpoint template code (that such applications can't interpret properly) getting in the way.

In the HTML Support folder, the master template is in a folder named the template name with *_master* (underline plus the word *master*) appended. The code that makes an .html file a Media Publisher template can be added later as described in "Use a Web Authoring Application to Create a Custom Template."

Building a Custom HTML Template

Create a new Media Publisher HTML template according to how you work best. If you want to design the look of your template in a Web authoring application first, and then add the code that enables it for Viewpoint Media Publisher, follow the steps below in "Use a Web Authoring Application to Create a Custom Template."

If you prefer to code directly in the .html file, follow the steps in "Edit the Code in an HTML Template to Create a Custom Design."

Use a Web Authoring Application to Create a Custom Template

As described above in "Use a Template Master as a Design Shortcut," you can start with a master file.

1 In a Web authoring application (Adobe GoLive is Viewpoint's recommended Web authoring tool), design the Web page you want to use as a template for your project.

2 When you are satisfied with your template design, open the resulting file in a text editor, such as Notepad. Launch a second instance of the text editor, and open **generic.html**.

3 In the new template file, add the HTML code and JavaScript that enables it for use with Media Publisher.

The following walks you through each copy and paste from generic.html to your new template.

- Copy (Ctrl + C) from before the `<html>` tag in generic.html:

```
<!-- Viewpoint Media Publisher scene description file. -->
<!-- Build ((VMP))BUILD((VMP)) -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

  Paste (Ctrl + V) above the `<html>` tag in your new template.

- Copy (Ctrl + C) from between `<head>` and `</head>` (head opening and closing tags) in generic.html:

```
((VMP))DESCRIPTION::The Generic template will integrate your VET scene
into a simple html page and create buttons for any animations in the
.mtx file which have the On="0" (do not run on loading) setting.
Additional information can be found in the accompanying Viewpoint Media
Publisher User Guide.

The files required by the Generic Template are kept in the
generic_resources folder and the template references this folder. When
the Publisher creates your HTML page it also creates a new folder at the
same location and copies over the files in the generic_resources folder.
The new folder is named the same as your html file and ends with
"_resources".

When you upload your files to your server, be sure to include the
htmlfilename_resources folder as it is required.((VMP))

<!-- Trigger.js is used to trigger the automaic download of the
Viewpoint Media Player by Netscape broswers. The code below points to
the trigger.js located in the generic_resources file, however you can
change this to point anywhere, so long as the trigger.js file is present
and available in that location. You can also use an absolute reference
e.g. "http://www.myurl.com/trigger.js". -->

<script language="JavaScript" src="((VMP))PATH((VMP))/trigger.js" >
</script>
<script language="javascript"
src="((VMP))PATH((VMP))/MTS3Interface.js"></script>
```

```
<script language="VBScript"
src="((VMP))PATH((VMP))/mtsAxDetect.vbs"></script>
<script language="javascript">
<!--

var a;   //This global variable is for the VET content
//-->
</script>

<!-- javascript from earlier versions -->
<script language="JavaScript">

<!--
var isIE4 = navigator.appName == "Microsoft Internet Explorer" &&
parseInt(navigator.appVersion.substring(0,1)) >= 4;
var isNav = (navigator.appName.indexOf("Netscape") != -1);
var isIE  = (navigator.appName.indexOf("Microsoft") != -1);
var agt=navigator.userAgent.toLowerCase();
var isWin = agt.indexOf('win') != -1;
var isMac = agt.indexOf('mac') != -1;



function triggeranimation(name) {a.TriggerAnim(name); }
function stopanimation   (name) {a.StopAnim(name); }
function startanimation (name)  {a.StartAnim(name);   }

//-->
</script>
```

Paste (Ctrl + V) between `<head>` and `</head>` (head opening and closing tags) in your new template.

> Tip: You can edit the text highlighted in gray to describe your custom template. The text you put there appears in Media Publisher's interface.

- Copy (Ctrl + C) from between `<body>` and `</body>` (body opening and closing tags) in generic.html:

```
<!-- FIRST OBJECT/EMBED -->
<script language="javascript">
   var alt = "none"
   a = new MTSPlugin("((VMP))FILENAME((VMP))", ((VMP))Plug-in
Width::400((VMP)), ((VMP))Plug-in Height::400((VMP)),
"((VMP))BROADCASTKEY((VMP))", alt, "GenieMinimumVersion=((VMP))VMP
GenieMin Version::3.0.3.80::VERSION((VMP))");
</script>

<form>
((VMP))FORMSDATA((VMP))
</form>
```

Paste (Ctrl + V) between `<body>` and `</body>` (body opening and closing tags) in your new template.

> Note: This is where Media Publisher embeds the VET scene. To find out about how to embed multiple scenes in a single Web page, refer to *About Viewpoint MTS3Interface.js* in the Media Player Docs folder. Double-click this file to open and read it in your default Web browser.

4 You can add in replaceable template items if you want to add input fields to the Media Publisher interface. For example:

```
BackgroundImage=((VMP))Background Image File Name((VMP))
```

((VMP)) indicates a replaceable template item. When you select a template, Media Publisher generates a field in the user interface for any replaceable item that allows input of a value. That value then replaces the replaceable template item in the .html file Media Publisher generates.

You can also specify default values using *::* (two colons) in the replaceable template item. For example:

```
BackgroundImage=((VMP))Background Image File Name::background_v.gif((VMP))
```

creates a user input field “Background Image File Name” with the default value *background_v.gif*.

Note: Each replaceable template item must be uniquely named and, therefore, cannot use the same name as an already existing item. If replaceable items use names of existing items, they do not display in the interface.

There is also a TYPE field that can be used with template items. Currently, the only supported type is VERSION. It can be used to convert version strings to hexidecimal version numbers used by Viewpoint Media Player (VMP). The following version element exists in all Media Publisher templates, and allows you to set a minimum version of VMP that supports your scene. (Users are automatically updated if an earlier version is detected.)

```
GenieMinimumVersion="((VMP))VMP Min Version::3.0.3.80::VERSION((VMP))"
```

The zoomview.html and zoomviewoptimized.html templates have two additional version elements, which are required to support ZoomView and Flash Compatibility*. They allow you to set a minimum version of VMP that supports scenes using ZoomView or Flash Compatibility*. (Users are automatically updated if an earlier version is detected.)

```
GenieMinimumVersion="((VMP))VMP Min Version::3.0.7.00::VERSION((VMP))"
HostMinimumVersion="((VMP))VMP Min Version::3.0.7.00::VERSION((VMP))"
ComponentMinimumVersion="((VMP))VMP Min Version::3.0.7.00::VERSION((VMP))"
```

Note: The following keywords are reserved:

((VMP))FILENAME((VMP))	Placeholder for the .mtx or .mtz filename.
((VMP))BUILD((VMP))	Placeholder for the Media Publisher build version.
((VMP))FORMSDATA((VMP))	Placeholder for animation buttons from the .mtx or .mtz file.
((VMP))BROADCASTKEY((VMP))	Placeholder for the broadcast key URL.
((VMP))PATH((VMP))	Placeholder for the resource files path.
((VMP))DESCRIPTION((VMP))	Placeholder for an empty string used to display template information.

5 Save your finished HTML template in the **HTML Support** folder.

6 Create a resource folder in HTML Support by making a copy of generic_resources and renaming it: your template name followed by *_resources* (underline and the word *resources*). Copy your skin files to this folder.

7 Test your new template in Media Publisher by generating a Web page. Follow the steps in “Creating VET-Enabled Web Pages.”

Edit the Code in an HTML Template to Create a Custom Design

If you prefer to code directly in HTML, follow these steps to create a custom template:

1 Make a copy of **generic.html** located in the **HTML Support** folder, and then save it to a new file name.

This template is selectable from the HTML Template menu in Media Publisher.

2 Create a resource folder in **HTML Support** by making a copy of generic_resources and renaming it: your template name followed by *_resources* (underline and the word *resources*).

3 In a text editor such as Notepad, add your HTML code to the new template. You can also add in replaceable template items if user customization is required. See step 4 of "Use a Web Authoring Application to Create a Custom Template" for instructions on using replaceable template items.

4 Add any skin files to the new template's resources folder, including logo images, background images, and stylesheet (.css) files.

5 Test the new template in Media Publisher, following the steps in "Creating VET-Enabled Web Pages."

Examples of Tags Generated by Media Publisher

Viewpoint Experience Technology (VET) Object/Embed Tag

Note: The MTS3Interface.js file has greatly reduced the amount of code that is embedded in the .html file, as compared to .html files generated by MTX 2 HTML.

Example of a VET Object/Embed Tag (Absolute Reference)

Following is an example of an object/embed tag with an absolute reference to the location of the Broadcast Key .txt file. In this case, broadcastkey.txt is is referenced in an absolute URL location. Highlighted areas show information entered into Media Publisher.

```
<!-- FIRST OBJECT/EMBED -->
<script language="javascript">
   var alt = "none"
   a = new MTSPlugin("bike_meter.mtx", 400, 400,
"http://bicycleaccessories.com/home/broadcastkey.txt", alt,
"GenieMinimumVersion=50332496");
</script>
```

Example of a VET Object/Embed Tag (Relative Reference)

Following is an example of an object/embed tag with a relative reference to the location of the Broadcast Key .txt file. In this case, broadcastkey.txt is in the same directory as the .html file. Highlighted areas show information entered into Media Publisher.

```
<!-- FIRST OBJECT/EMBED -->
<script language="javascript">
   var alt = "none"
   a = new MTSPlugin("bike_meter.mtx", 400, 400, "BroadcastKey.txt", alt,
"GenieMinimumVersion=50332496");
</script>
```

Media Publisher and VET Animation Tags

Viewpoint Media Publisher looks for any VET animation tags in the .mtx/.mtz file and automatically generates buttons in the HTML page to trigger those animations using JavaScript.

Example of a VET Animation Tag in an .mtx File

Interactor

This interactor element in `<MTSInstance>` references a texture animation that creates the effect of activating the LCD display for the bicycle meter.

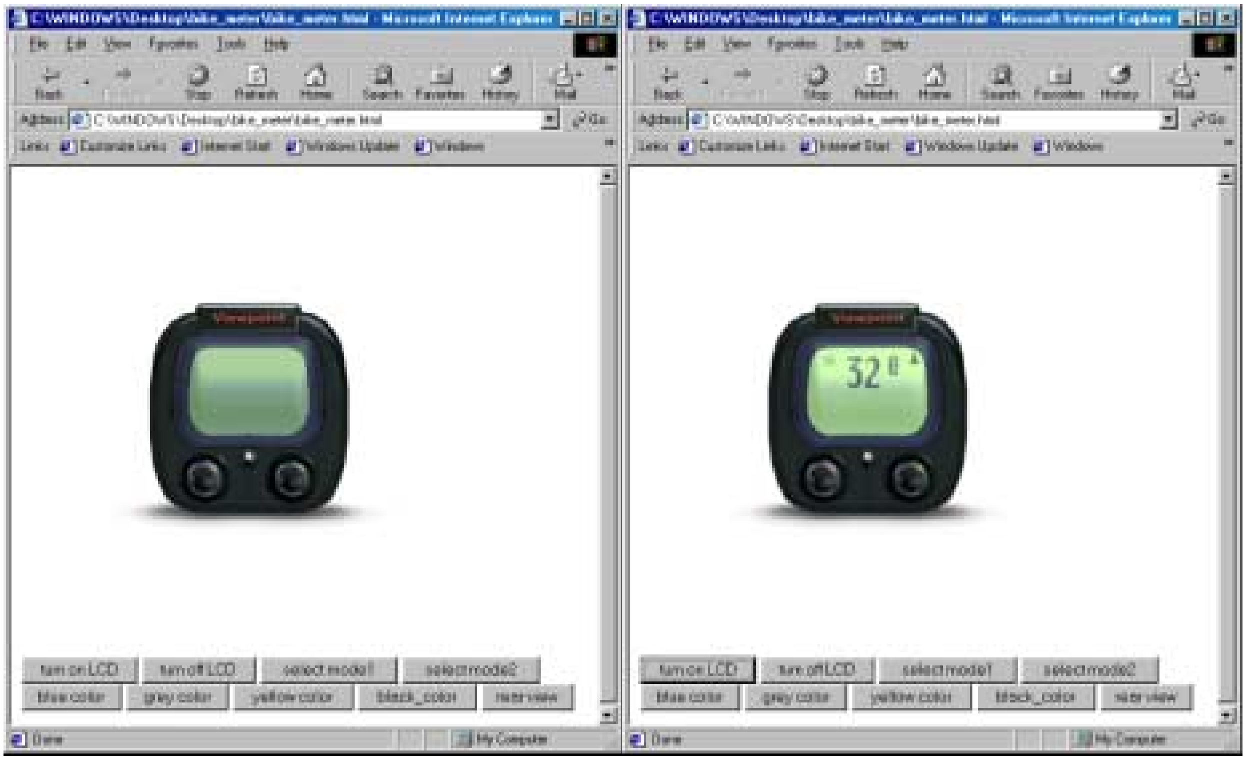
```
<OnClick Action="Trigger" ID="0" Target="turn on LCD"/>
```

Keyframe Animator

This texture animation, when triggered by OnClick illustrated above, replaces the texture on the face of the meter with a texture that looks like an activated LCD display.

```
<MTSTimeElem Name="turn_on_LCD"  Type="Keyframe"  On="0" >
   <Target Name="bike_meter_TEXTURE_2"  Property="pixl" Timeline="T1" />
   <Time>   0    .5    1                          1.25       </Time>
   <Timeline Name="T1" Type="Texture">
            *    *     [ bike_meter_TEXTURE_15 ]   [ bike_meter_TEXTURE_16 ]
   </Timeline>
</MTSTimeElem>
```

Bicycle Meter Scene Embedded in an HTML Page Created by Media Publisher


C:\WINDOWS\Desktop\bike_meter\bike_meter.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail
Address C:\WINDOWS\Desktop\bike_meter\bike_meter.html
Go
Links Customize Links Internet Start Windows Update Windows
Viewpoint
32
turn on LCD
turn off LCD
select mode1
select mode2
blue color
grey color
yellow color
black_color
rear view
Done
My Computer

Media Publisher automatically generates buttons that trigger the animations in a scene. In this example, a texture animation creates the effect of an LCD display being activated.

Example of JavaScript and Form Tags Created in an .html File

```
<!-- Trigger.js is used to trigger the automatic download of the Viewpoint
Media Player by Netscape broswers. The code below points to the trigger.js
located in the generic_resources file, however you can change this to point
anywhere, so long as the trigger.js file is present and available in that
location. You can also use an absolute reference e.g.
"http://www.myurl.com/trigger.js". -->

<script language="JavaScript" src="bike_meter_Resources/trigger.js" >
</script>
<script language="javascript"
src="bike_meter_Resources/MTS3Interface.js"></script>
<script language="VBScript"
src="bike_meter_Resources/mtsAxDetect.vbs"></script>
<script language="javascript">
<!--

var a;    //This global variable is for the VET content
//-->
</script>

<!-- javascript from earlier versions -->
<script language="JavaScript">

<!--
var isIE4 = navigator.appName == "Microsoft Internet Explorer" &&
parseInt(navigator.appVersion.substring(0,1)) >= 4;
var isNav = (navigator.appName.indexOf("Netscape") != -1);
var isIE  = (navigator.appName.indexOf("Microsoft") != -1);
var agt=navigator.userAgent.toLowerCase();
var isWin = agt.indexOf('win') != -1;
var isMac = agt.indexOf('mac') != -1;



function triggeranimation(name) {a.TriggerAnim(name); }
function stopanimation   (name) {a.StopAnim(name); }
function startanimation (name)  {a.StartAnim(name);   }

//-->
</script>


</head>

<body>
<!-- FIRST OBJECT/EMBED -->
<script language="javascript">
   var alt = "none"
   a = new MTSPlugin("bike_meter.mtx", 400, 400, "BroadcastKey.txt", alt,
"GenieMinimumVersion=50332496");
</script>
```

```
<form>
<input name="turn_on_LCD" type=button  value="turn_on_LCD"
   onclick="triggeranimation('turn_on_LCD')">
<input name="turn_off_LCD"   type=button  value="turn_off_LCD"
   onclick="triggeranimation('turn_off_LCD')">
<input name="select_mode1"   type=button  value="select_mode1"
   onclick="triggeranimation('select_mode1')">
<input name="select_mode2"   type=button  value="select_mode2"
   onclick="triggeranimation('select_mode2')">
<input name="blue_color"  type=button  value="blue_color"
   onclick="triggeranimation('blue_color')">
<input name="grey_color"  type=button  value="grey_color"
   onclick="triggeranimation('grey_color')">
<input name="yellow_color"   type=button  value="yellow_color"
   onclick="triggeranimation('yellow_color')">

<input name="black_color" type=button  value="black_color"
   onclick="triggeranimation('black_color')">
<input name="rear_view"   type=button  value="rear_view"
   onclick="triggeranimation('rear_view')">

</form>
```

Appendix: Registering MIME Types

Registering MIME types (also known as content or media types) is a very important step to carry out before deploying VET content on a Web server. Viewpoint recommends that you register **.mtx**, **.mtz**, **.mts**, and **.mzv** as MIME types.

Following are instructions on how to use a text editor on your Apache system to edit the 'mime.types' file and add MIME types for Viewpoint.

Important: Be sure to use spaces exactly as shown in all the syntax provided here.

1 Login as root.

2 Locate an editor to use:

While there can be many different text editors on a system, 'joe' is usually on all UNIX servers (others include 'vi', 'pico', or nano'). Use 'which joe' or 'which vi' to confirm the existence of the text editor. If it returns a path, like /usr/local/bin/joe, then you’ve got it on your system. 'vi' should be on every system, but 'joe' is easier to use.

3 Modify the mime.types file. There are two approaches to changing the mime.types file. To determine which method to use, do the following:

- Open a shell session.
- Type 'apache -l' for Apache version 1.3.9 or earlier. Or, type 'httpd -l' for Apache version 1.3.10 or later.

If 'mod_mime.c' is among the values returned, follow the instructions in “Use the AddType Directive” listed below. If you do not see 'mod_mime.c', then you must manually edit the mime.types file (see “Manually Edit the mime.types File” for instructions).

Manually Edit the mime.types File

1 Use 'find / -name mime.types' to find the mime.types file. You will probably see the mime.types file in several places, including documentation. In older versions of Apache, it might be under the /etc/apache directory; in newer versions, it should be under /usr/local/apache/conf. You should select the primary mime.types file to edit.

2 Type 'joe PREFIX/mime.types', where PREFIX is the path to the mime.types file found in step 1.

3 In the application area, add 'application/metastream mtx mts mtz mzv'.

4 Save the file (in joe, hit ctrl-k, and then the letter 'x'). You should see a message indicating that the mime.types file was saved.

5 Type 'apachectl restart'. You should see messages indicating that Apache and/or httpd were restarted.

Use the AddType Directive

Note: These instructions can be found at http://httpd.apache.org/docs/mod/mod_mime.html - addtype.

AddType Directive

Syntax	AddType MIME-type extension [extension ...]
Context	server config, virtual host, directory, .htaccess
Override	FileInfo
Status	Base
Module	mod_mime

The AddType directive maps the given filename extensions onto the specified content type. MIME-type is the MIME type to use for filenames containing extension. This mapping is added to any already in force, overriding any mappings that already exist for the same extension. This directive can be used to add mappings not listed in the MIME types file (see the TypesConfig directive). For example:

```
AddType application/metastream mtx mts mtz mzv
```

It is recommended that new MIME types be added using the AddType directive rather than changing the TypesConfig file.

Note: Unlike the NCSA httpd, this directive cannot be used to set the type of particular files.

Glossary

Broadcast Key A unique alphanumeric string issued by Viewpoint Corporation to companies or individuals licensed to broadcast Viewpoint Experience Technology (VET) content. The string is stored in a text (.txt) file that is referenced by VET-enabled Web pages. VET content without a Broadcast Key displays with a watermark (the name VIEWPOINT).

MTS A Viewpoint Media File that contains a compressed collection of all the resources referenced by the .mtx/.mtz file (such as geometry, Lightmaps, and texture maps) for the creation of a 3D scene. A published Viewpoint Experience Technology (VET) scene consists of two Viewpoint Media Files, an .mts file and an .mtx/.mtz file.

MTX An XML-based Viewpoint Media 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 references the .mts file.

MTZ A compressed form of an .mtx file that the preferred Viewpoint Media File format for Web-enabled Viewpoint content. Sometimes animations in the .mtx file are extensive, making the file size large. Compressing large .mtx files enables fast downloading of Viewpoint Experience Technology scenes.

MZV A file format for compressed image tiles (sections of a high-resolution image) used by the ZoomView component of Viewpoint Media Player.

skin In the case of Viewpoint Media Publisher, a skin is the overall look of a Web page as determined by design elements, such as a stylesheet (.css file), background color, background image, and logo image. Uses can customize template skins to meet their design needs.