ImageGen

REFERENCE MANUAL

Date
November 2013

Creator
Douglas J. Robar, Percipient Studios.

Table of Contents

Installing ImageGen

INSTALLING

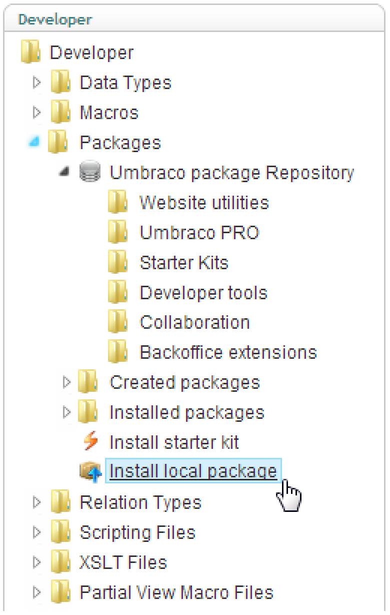
Package Installation for Umbraco Websites

1. Open the Umbraco administration interface and go to the *Developer* section. Expand the *Packages* area.

2. Browse the *Website utilities* section and select *ImageGen*.

   Or, download the *ImageGen 2.9* package zip file to your local drive (do not expand the zip file) and select the *Install local package* menu.

3. Follow the on-screen instructions to complete the installation.



Manual Installation for .NET Websites

To manually install ImageGen in any .NET-based website, download and expand the files from the *ImageGen 2.9* package zip file. Then:

1. Copy `ImageGen.dll` to the `/bin` folder

2. Copy `ImageGen.ashx` to any folder in your site (these instructions assume the `/` root of your site)

3. Copy the sample `ImageGen.config` file to any folder in your site (we will assume the `/config` folder)

4. Update the `/web.config` file by adding the following entries (in bold) to the `<configuration>` section:

```
<configuration>
   …
   <configSections>
       <section name="ImageGenConfiguration" type="ImageGen.ImageGenConfigurationHandler,ImageGen" />
   </configSections>

   <ImageGenConfiguration configSource="config\ImageGen.config" />
   …
</configuration>
```

UPDATING FROM PREVIOUS VERSIONS

Package Installation for Umbraco Websites

1. Download the *ImageGen 2.9 Updater* package zip file to your local drive (do not expand the zip file).
2. Open the Umbraco administration interface and go to the *Developer* section. Expand the *Packages* area.
3. Select the *Install local package* menu.
4. Follow the on-screen instructions to complete the installation.
5. Delete all `cached` folders on the site. New cached folders will be created as needed.

Manual Installation for .NET Websites

1. Download the *ImageGen 2.9 Updater* package zip file to your local drive.
2. Unzip the updater file.
3. Replace the `ImageGen.ashx` and `/bin/ImageGen.dll` files in your site
4. Delete all `cached` folders on the site. New cached folders will be created as needed.

ACTIVATING PROFESSIONAL FEATURES

In order to activate the *ImageGen Professional* features on a live domain, first purchase a registration key for the domain(s) from www.percipientstudios.com, and then update the `/config/ImageGen.config` file to include the registration key(s), similar to that shown below:

```
<ImageGenConfiguration>
   <Registration>
       <Key domain="example.com">C90FE21F1447D822BB4D834BC46F7D89A1376272</Key>
       <Key domain="sample.com">55FD8FD97A199AB3082BC7DC11AE5F743B092C55</Key>
   </Registration>
   …
</ImageGenConfiguration>
```

Note that all *ImageGen Professional* features enabled when running from `http://localhost` or any `http://*.local` domain. This allows you to use the professional features during development or to try the professional features before buying a registration key for your or your client's domain.

ImageGen Cheatsheet

`Align``=center`	Options: `left, center, right, near, far`
`AllowUpSizing``=false`	Options: `true, false`
`AltBrowser``=Browser:IE;MajorVersion:^[1-6]$`	
`AltImage``=/media/headshot.jpg`	(path and image filename)
`AntiAlias``=false`	Options: true, false
`BgColor``= FFC316`	(a hex value)
`Border``=2`	(an integer)
`BorderColor``= 30291F`	(a hex value)
`Class``=headshot`	(a named class in imagegen.config file)
`ColorMode``=Grayscale`	Options: `color, grayscale, sepia`
`Compression``=75`	(an integer, 0-100)
`Constrain``=true`	Options: `true, false`
`Crop``=Resize`	Options: `resize, noresize` or `x,y,width,height`
`Flip``=XY`	Options: `x, y, xy`
`Font``=Verdana`	(installed font name, or Font=/fonts/myfont.otf)
`FontColor``=000033`	(a hex color value)
`FontSize``=72`	(a number)
`FontStyle``=Bold%2BItalic`	Options: `regular, bold, italic, underline, strikeout`
`Format``=PNG`	Options: `JPEG, JPG, GIF, PNG, BMP, TIFF, TIF`
`Height``=75`	(an integer)
`Image``=/media/beautiful.jpg`	(path and image filename)
`LineHeight``=40`	(a number)
`MaxHeight``=800`	(an integer)
`MaxWidth``=800`	(an integer)
`NoCache``=false`	Options: `true, false`
`OverlayImage``=/images/watermark.png`	(path and image filename)
`OverlayMargin``=8`	(an integer)
`Pad``=true`	Options: `true, false`
`Rotate``=30`	(a number)
`Text``=Super%20Cool!`	(html-encoded and then url-encoded text)
`Transparent``=false`	Options: `true, false`
`Valign``=top`	Options: `top, middle, bottom, near, far`
`Version`	(must be the only parameter)
`Width``=200`	(an integer)

ImageGen Reference

ALIGN

The horizontal alignment/position of text in a text-only image if `Width` is also specified. The horizontal position of text or an overlay image atop an image, even if `Width` is not specified.

While most languages are read left-to-right, Arabic, Hebrew and others are displayed right-to-left. Using `Near` and `Far` will align text appropriately based on the natural direction of the Unicode text being rendered. The edge nearest the start of a line is `Near` and `Far` is the ending edge.

Class Notation:	`<Align>Left</Align>`
Options:	`Left, Center, Right, Near, Far`
Default:	`Left`
Also See:	`VAlign`

ALLOWUPSIZING

ImageGen Professional

Allow or disallow an image to be resized larger than its original width and height. If `AllowUpSizing=false` the image produced by *ImageGen* will not exceed its original width or height, even if the `Width` or `Height` parameters requested are greater than the dimensions of the original image.

Class Notation:	`<AllowUpsizing>True</AllowUpsizing>`
Options:	`True, False`
Default:	`True`
Also See:	`MaxHeight, MaxWidth`

ALTBROWSER

ImageGen Professional

Display the `AltImage` rather than the `Image` if the user's browser matches the specified `AltBrowser` criteria. This is especially useful for displaying a .GIF image to older browsers but a .PNG to newer browsers. `AltBrowser` has no effect if `AltImage` is not also specified. Limited regular expression matching is supported. Separate multiple `AltBrowser` parameters with a semi-colon. Note that using `AltBrowser` will slow the performance of *ImageGen* slightly.

Class Notation:	`<AltBrowser>Browser:IE</AltBrowser>`
Examples:	`Browser:Gecko`
	`Browser:IE;MajorVersion:^[1-6]$` `Type:IE5;Platform:/macppc/i`

Example parameters:

	Firefox 2.0	Chrome 6.0	IE6
`Id`	mozillafirefox	safari1plus	ie6to9
`Type`	Firefox	Safari	IE6
`Browser`	Gecko	AppleWebKit	IE
`Version`	1.8	534	6.0
`Majorversion`	1	5	6
`MinorVersion`	0.8	0.34	0
`Platform`	WinXP	WinXP	WinNT
`EcmaScriptVersion`	1.5	1.5	1.2
`Cookies`	True	True	True
`VBScript`	False	True	True
`JavaScript`	True	True	True

For a list of properties, see the HttpBrowserCapabilities class at
http://msdn.microsoft.com/en-us/library/system.web.httpbrowsercapabilities.aspx

ALTIMAGE

If `Image` is not found on the server, `AltImage` is used instead. This is especially useful for displaying a generic “photo not available” image in a staff directory list, for instance. The `AltImage` can also be activated by using the `AltBrowser` parameter. Specify the path and filename on the web server to an alternate image.

`AltImage` must be a local file. `ImageBaseDir` will be applied only if it doesn’t contain `://`.

Class Notation:	`<AltImage>/photos/waterfall.png</AltImage>`
Also See:	`AltBrowser`, `Class` (`ImageBaseDir` property)

ANTIALIAS

Remove “jaggies” around the edges of rendered text. The `BgColor` specifies the color against which the text is anti-aliased.

Class Notation:	`<AntiAlias>True</AntiAlias>`
Options:	`True`, `False`
Default:	`True`
Also See:	`BgColor, Format, Transparent`

BGCOLOR

The background color to place behind the text or image. `BgColor` is only visible around text if `Transparent=false` or `Format=` is not GIF or PNG. `BgColor` is only visible around an image if `Pad=True` or the image is rotated. Colors can be specified by their hex value (without the # symbol), or by name. The named colors can be found at http://msdn.microsoft.com/en-us/library/system.drawing.knowncolor.aspx

Class Notation:	`<BgColor>FFFFFF</BgColor>`
Default:	`white` (`FFFFFF`).
Also See:	`AntiAlias, Transparent`

BORDER

The width of a border to place around the edge of an image or text graphic. A negative border will reduce the image on all sides by the border amount. With a negative border the overall image dimensions are reduced by the border amount, unless `Width` or `Height` is specified, in which case the final dimensions will be those specified but with the image cropped by the negative border amount.

Class Notation:	`<Border>0</Border>`
Default:	`0`

BORDERCOLOR

The border color. Colors can be specified by their hex value (without the # symbol), or by name. The named colors can be found at http://msdn.microsoft.com/en-us/library/system.drawing.knowncolor.aspx

Class Notation:	`<BorderColor>000000</BorderColor>`
Default:	`black` (`000000`)
Also See:	`Border`

CLASS

ImageGen Professional

A `Class` is a group of parameters defined in the `ImageGen.config` file. By pre-defining a group of parameters you only need reference the `Class` on the querystring without revealing all the details. This simplifies site maintenance with centralized settings, shortens URLs, and provides increased security. All querystring parameters can be specified in a `Class`. Note that some parameters can *only* be specified in a `Class`. See below for details.

An Example

Without classes, you would specify every parameter on the querystring, such as:

```
ImageGen.ashx?image=/photos/big.jpg ↩
   &overlayImage=/photos/watermark.png&width=200&height=200&constrain=true
```

But with the following named class defined in the ImageGen.config file:

```
<ImageGenConfiguration>
   …
   <Class Name="Thumbnail" OverridesQueryString="true">
       <Constrain>true</Constrain>
       <ImageBaseDir>/photos</ImageBaseDir>
       <OverlayImage>watermark.png</OverlayImage>
       <Height>200</Height>
       <Width>200</Width>
   </Class>
</ImageGenConfiguration>
```

You would only need to specify the source image and the class to apply:

```
/ImageGen.asxh?image=big.jpg&class=thumbnail
```

Without classes the command string is not only much longer, it reveals important information about the image (such as its original location at /media/big.jpg). A malicious website visitor could use this information to request the original image at full size without the watermark. With classes and the `OverridesQueryString="true"` setting, the location of the original file is not revealed, the watermark cannot be removed, nor can the size be altered.

Class Structure

Every Class will have a Name or Folder attribute. A Class declaration may also have an optional attribute to ensure the class settings cannot be altered by setting different parameter values on the querystring. Within the Class, parameters are defined as elements.

To avoid redundancy, classes can inherit the settings of another Class.

TIP: To unset a parameter in an inherited class that is set in the parent class, specify an empty parameter, such as `<OverlayImage></OverlayImage>`.

Named Classes

The parameters of a named class are activated when the class name is included in the querystring
`imagegen.ashx?image=photo.png&class=thumbnail`

If the class specified on the querystring does not exist, *ImageGen* will behave as though no class were specified.

The Special "Default" Named Class

If `<Class Name="default">` exists, its settings will be applied automatically whenever the `class=` parameter is omitted from the querystring, or if the requested querystring `class=` does not exist.

TIP: It is best practice to specify strict values for all parameters in the default class. This effectively limits *ImageGen* to a minimum behavior unless a named class is explicitly requested, which should also have very specific parameters. This best protects your media assets.

Folder Classes

Folder classes are automatically activated based on the location of the source image. Like the default class, folder classes do not require the `class=` on the querystring. A folder class is applied when the first characters of the `Folder=` property exactly match the first characters of the `image=` querystring parameter.

You may specify a single image (for example, `Folder="logo.jpg"`) or an entire folder and all images contained in that folder (for example, `Folder="/screenshots"`). Folder classes do not apply to sub-folders; you must specify a folder class for each folder (though each folder can inherit from a base class).

Simple regular expressions can be used to match folder classes. For instance, `Folder="[clients|partners]/logos"` would match images in both the `clients/logos` folder and the `partners/logos` folder. Please note that a leading forward slash indicates a folder and will disable regex evaluation.

TIP: A named class referenced in the `class=` querystring parameter takes precedence over a folder class that would otherwise have been activated if the class had not been specified; folder classes take precedence over the default class.

Override Querystring Parameters

The `OverridesQueryString` class attribute ensures that any settings specified in the class will be applied, even if different values are specified on the querystring. If `OverridesQueryString` is set to false, parameter values on the querystring will be used even if the same parameter is also specified in the class. The default value is `true` if not specified.

TIP: Avoid setting `OverridesQueryString` to false, because this eliminates protection against potentially malicious users.

Inherit from Class

The `InheritFromClass` element lets you inherit or cascade settings from the specified class, allowing you to override any inherited parameter without repeating all the parameters of the parent class.

Settings Only Available in a Class

ImageBaseDir

The `ImageBaseDir` parameter element specifies a path fragment that is silently prepended to `Image=`, `AltImage=`, and `OverlayImage=` querystring parameters. This prevents the full path of the image from being revealed on the querystring, as well as shortening the querystring.

TIP: You can also use remote images but not advertise the fact by including the remote domain and image path. For instance, `<ImageBaseDir>http://www.example.com/logos</ImageBaseDir>`.

CachingTimeSpan

The `CachingTimeSpan` parameter element sets the number of seconds the website visitor's browser should cache the image locally before requesting a new version from the server. Unless specified, there is no client-side caching, which is equivalent to `<CachingTimeSpan>0</CachingTimeSpan>`. Server-side caching with ETags and 304-Not Modified responses is always enabled, even with *ImageGen Basic*.

TIP: For resized images that change infrequently, set a large `CachingTimeSpan` value for the class. This will maximize site responsiveness and minimize requests to the server. This is particularly effective with large photo galleries but almost all classes will benefit from some client-side caching.

Font Style Offsets

Because the font metrics available to ASP.NET are not adequate to ensure pixel-perfect baseline alignment when switching amongst fonts in a single line of text, you may need to manually assign and offset for Bold, Italic, and/or BoldItalic fonts within the class. A positive offset will move that font style up, a negative offset shifts text in the font style down.

Offsets are only active when using `&class=` on the querystring, but not when the class name is embedded within the `&text=` parameter. That is, `&text=hello,<myClass>world</myClass>` won't adjust baselines, while `&class=myclass&text=hello, <b>world</b>` would apply baseline offsets.

TIP: You may need differing offset values, even for the same font, for various font sizes.

```
<Class Name="LargeHeadingGreen">
    <Font>AvantGardeCE</Font>
    <FontSize>53</FontSize>
    <FontColor>c6ff00</FontColor>
    <BoldOffset>0</BoldOffset>
    <ItalicOffset>-2</ItalicOffset>
    <BoldItalicOffset>1</BoldItalicOffset>
</Class>
```

COLORMODE

ImageGen Professional

Convert a color image to grayscale. Note that changing an image's color is a processor-intensive operation, particularly for larger images.

Class Notation:	`<ColorMode>Color</ColorMode>`
Options:	`Color`, `Grayscale`, `Sepia`
Default:	`Color`

COMPRESSION

For JPEG images, `Compression` specifies the file size vs. quality trade-off. `0` provides minimum file size (and minimum quality) while `100` provides maximum quality (and maximum file size). The default value produces a very good quality image of reasonable size.

Class Notation:	`<Compressionr>90</Compression>`
Options:	`0` to `100`
Default:	`90`

CONSTRAIN

Retain the proportions (aspect ratio) of the original image when resizing to fit within the specified `Width` and `Height`. Thus, one of the dimensions of the resized image will be exactly that specified by the `Width` or `Height` parameter and its other dimension will be smaller. If `Constrain=false`, the resized image will fill the specified `Width` and `Height` without regard for the original image's proportions, potentially distorting it. `Constrain` is only used when both `Width` and `Height` are specified and `Pad` is false; otherwise it is ignored.

Class Notation:	`<Constrain>False</Constrain>`
Options:	`True`, `False`
Default:	`False`
Also See:	`Pad`

CROP

ImageGen Professional

Crop the image to the specified `Width` and `Height`. There are three cropping modes to choose from.

If `Crop=resize` is specified, the image is first resized and any portion of the image that extends beyond the specified dimensions is then cropped off. This results in an image of the exact dimensions specified by the `Width` and `Height` parameters and reveals the maximum amount of the original image.

If `Crop=noresize` is specified, the original, un-resized image is immediately cropped to the `Width` and `Height` dimensions specified and no further resizing takes place. The portion of the original image that remains visible is based on the `Align` and `VAlign` settings.

Alternatively, `Crop=10,50,200,200` allows you to specify the top-left coordinates (`X,Y`) and the area (`W,H`) to retain. This cropped portion of the image is then resized to the `Width` and `Height` specified. Cropping with the x,y,w,h option retains aspect ratio of the image if only `Width` or `Height` is specified (if both `Width` and `Height` are specified you will need to be careful to get aspect ratio correct). In this mode, any `Align` and `VAlign` settings are ignored insofar as they might otherwise affect cropping.

Class Notation:	`<Crop>Resize</Crop>`
Options:	`Resize, Noresize, X,Y,W,H`
Also See:	`Constrain, Pad, Align, VAlign`

FLIP

ImageGen Professional

Mirror the image horizontally, vertically, or both.

Class Notation:	`<Flip>X</Flip>`
Options:	`X, Y, XY`
Also See:	`Rotate`

FONT

The font used for rendering text. Specify the font name of any font installed on your web server, or the path and filename of a TrueType or Open Type font located on your web server. If the specified font cannot be found, the default font (Arial) is used.

Class Notation:	`<Font>Arial</Font>`	
Default:	`Arial`	
Examples:	`Font=Verdana`	(for installed fonts)
	`Font=/fonts/myfont.otf`	(for font files not installed on the server)
Also See:	`Class, FontStyle, Text`	

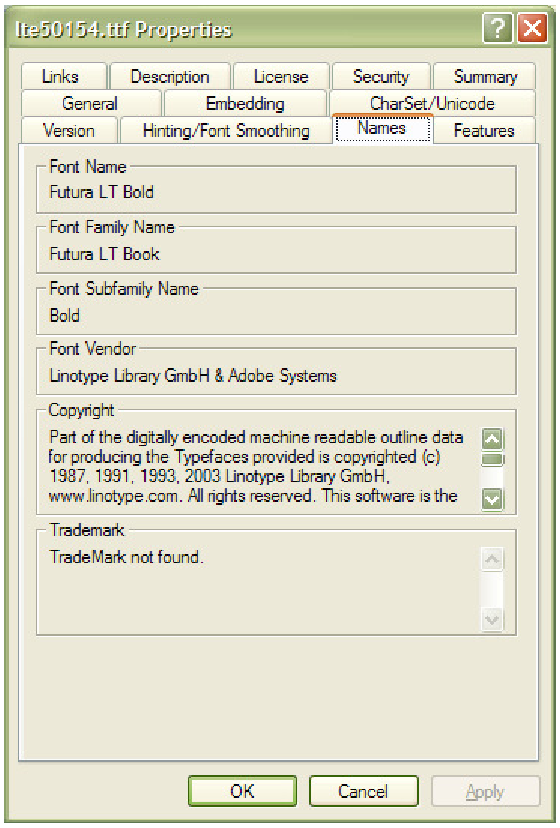
Note that Open Type support is limited to those fonts with TrueType outlines rather than Postscript outlines. This is a limitation of the .NET Framework.

Using Uninstalled Fonts

When an individual/uninstalled font file is specified, that font file contains a single font style (regular, bold, italic, bold-italic). Whereas when a font is installed on the server all available font styles are available. Therefore, if you specify a font by filename you must also specify the `FontStyle` that matches the style contained in the font file. A mismatch will prevent the font from displaying properly.

```
ImageGen.ashx?Text=Hello&Font=/fonts/lte50154.ttf&FontStyle=bold
```

Detailed font information can be found using the free *Microsoft Font Properties Extension* for x86 operating systems, as shown below.



Font Name
Futura LT Bold

Font Family Name
Futura LT Book

Font Subfamily Name
Bold

Font Vendor
Linotype Library GmbH & Adobe Systems

Copyright
Part of the digitally encoded machine readable outline data for producing the Typefaces provided is copyrighted (c) 1987, 1991, 1993, 2003 Linotype Library GmbH, www.linotype.com. All rights reserved. This software is the

Trademark
TradeMark not found.

Using Uninstalled Fonts – *ImageGen Professional*

With *ImageGen Professional*, you can further define a collection of individual/uninstalled font files, providing the same capabilities as you get when using fonts that are installed on the server.

Update the `ImageGen.config` file with a `<Font>` definition, giving it a descriptive name (do not use a period in the name), and defining the path to the file for each style.

```
    <Font Name="CorporateFont">
        <Filename Style="regular">/fonts/AGW_CE.ttf</Filename>
        <Filename Style="bold">/fonts/AGD_CE.ttf</Filename>
        <Filename Style="italic">/fonts/LTe52012.ttf</Filename>
        <Filename Style="bold, italic">/fonts/LTe52014.ttf</Filename>
    </Font>
```

Please note that the `Style` attribute's value (`regular`, `bold`, etc.) must be typed in lower-case. As with any uninstalled font, be sure that the font file specified supports the font style it is associated with.

TIP: In some situations, a font file may have an obscure or non-standard style definition. This can often be handled by specifying the optional `SubFamily` attribute as shown in *Microsoft Font Properties Extension*. In such cases, determining the right combination of `Style` and `SubFamily` attributes is vital.

```
<Filename Style="italic" SubFamily="Black">/fonts/snellk.ttf</Filename>
```

FONTCOLOR

The rendered text's color. Colors can be specified by their hex value (without the # symbol), or by name. The named colors can be found at http://msdn.microsoft.com/en-us/library/system.drawing.knowncolor.aspx

Class Notation:	`<FontColor>000000</FontColor>`
Default:	`black` (`000000`)

FONTSIZE

The size of rendered text, in points.

Class Notation:	`<FontSize>32</FontSize>`
Default:	`32`
Also See:	`Text`

FONTSTYLE

The rendered text should be shown with this style. Any combination of `FontStyle` can be used, but must be separated by plus signs (`%2B` is the URL-escaped plus sign for use with querystrings). Not all styles are supported by all fonts, especially those fonts which are not installed on the web server.

Class Notation:	`<FontStyle>Bold+Italic</FontStyle>`
Options:	`Regular, Bold, Italic, Underline, Strikeout`
Default:	`Regular`

Examples:	`FontStyle=Bold`
	`FontStyle=Bold%2BItalic%2BUnderline`

Also see:	`Class, Font, Text`

FORMAT

The image format for the rendered or resized image. If the `format` is not specified, text-only images will be transparent GIF (note the `BgColor` defaults to white), and resized images will be in the same format as the original image.

Class Notation:	`<Format>GIF</Format>`
Options:	`JPEG, JPG, GIF, PNG, BMP, TIFF, TIF`
Default:	The format of the original image when resizing, or `GIF` for text-only graphics

TIFF support

Because TIFF is not viewable in a browser it will be downloaded instead. TIFF images using a CMYK colorspace will be converted to RGB when resized, which may shift the colors slightly.

Supported TIFF input file formats include:

8-bits per pixel (16-bit images are not supported)
RGB or CMYK color
(Include ICC/ICM information when using CMYK to avoid color shifts)
LZW compression (ZIP and JPEG compression is not supported)
Interleaved pixel order
IBM PC or Macintosh byte order
Image Pyramid
Layers
RLE or ZIP layer compression

TIFF output file format is:

8-bits per pixel
RGB color
LZW compression

HEIGHT

The height of the created or resized image. The image will resize proportionally to the desired height unless `Width` is also specified. If both `Width` and `Height` are specified, the image is resized to those dimensions exactly. If `Constrain` or `Pad` is specified in addition to `Width` and `Height`, the image is resized proportionally and/or padded to fill any extra space.

Class Notation:	`<Height>100</Height>`
Default:	The height of the original image when resizing, or to the height required to contain the specified text graphic
Also See:	`Constrain`, `Pad`, `Width`

IMAGE

The full path and filename on the web server to an image to resize, or URL of remote image to resize. *ImageGen* can also automatically select the `{first}` image or a `{random}` image from a folder of images.

Class Notation:	`<Image>portrait.jpg</Image>`

Examples:	`/media/big.gif`
	`http://example.com/media/photo.jpg`
	`/images/screenshots/{first}`
	`/images/{random}`

Also See:	`AltBrowser`, `Class` (`ImageBaseDir` property)

LINEHEIGHT

ImageGen Professional

The height of a line (leading) of text, in points. If not specified the leading is approximately 1.2X the `FontSize`. A setting of `-1` forces the default setting. A setting of `0` allows lines of text to write on top of themselves.

Class Notation:	`<LineHeight>40</LineHeight>`
Default:	Equal to the current FontSize
Also See:	`FontSize`

MAXHEIGHT

ImageGen Professional

The maximum height of a rendered or resized image. If the `Height` parameter is larger than the specified `MaxHeight` value, the `Height` will be treated as though it were equal to the `MaxHeight` setting. This is an extremely useful parameter for use in classes to avoid the creation of inordinately large images.

Class Notation:	`<MaxHeight>800</MaxHeight>`
Also See:	`AllowUpsizing, Class`

MAXWIDTH

ImageGen Professional

The maximum width of any image. If the `Width` parameter is larger than the specified `MaxWidth` value, the `Width` will be treated as though it were equal to the `MaxWidth` setting. This is an extremely useful parameter for use in classes to avoid the creation of inordinately large images.

Class Notation:	`<MaxWidth>800</MaxWidth>`
Also See:	`AllowUpsizing, Class`

NOCACHE

Disregard any previously-cached image, recreate the image and cache the new image to disk.

Useful for debugging. This setting should not be used in production as it will create new images with every request, slowing performance and using more server resources than necessary.

Note: there is no way to disable the writing of cached images to disk.

Class Notation:	`<NoCache>False</NoCache>`
Options:	`True`, `False`
Default:	`False`

OVERLAYIMAGE

ImageGen Professional

The full path and filename to an image on the web server to overlay on top of the `Image` or `Text`. As with text atop images, `Align` and `VAlign` parameters specify the position of the `OverlayImage` with respect to the underlying image. Best results are achieved with transparent PNG overlay images, utilizing the alpha transparency channel.

Note that `OverlayImage`'s are neither resized nor rotated, but are placed at their original size on top of the already-resized `Image`. Therefore, you may want several different sizes of your watermark or logo for use with the various sizes of images you create with *ImageGen*.

Class Notation:	`<OverlayImage>watermark.png</OverlayImage>`
Also See:	`OverlayMargin, ImageBaseDir, Class`

OVERLAYMARGIN

ImageGen Professional

The distance the `OverlayImage` will appear from the edge specified by the `Align` and `VAlign` parameters.

Class Notation:	`<OverlayMargin>5</OverlayMargin>`
Also See:	`Align, VAlign, OverlayImage`

PAD

After resizing the image, fill any leftover area with the `BgColor` and place the (proportionally) resized image in the center of the available space. `Pad` is only useful when both `Width` and `Height` are specified; otherwise it is ignored.

Class Notation:	`<Pad>False</Pad>`
Options:	`True, False`
Default:	`False`
Also See:	`BgColor, Constrain, Crop, Transparent`

ROTATE

ImageGen Professional

The number of degrees to rotate the rendered text or image in a clockwise direction. As the image is rotated, the `BgColor` will fill any unused space. Note that the `OverlayImage`, if specified, is not rotated, though `Text` is rotated, even if placed on top of an image.

Class Notation:	`<Rotate>90</Rotate>`
Default:	`0`
Also See:	`BgColor, Transparent`

TEXT

ImageGen Professional for some features

The text to be rendered. If neither `Width` nor `Height` is specified the image will be as wide and tall as necessary to contain the text. If the `Text` and `FontSize` parameters generate text too large for the specified `Width` and `Height`, no text will appear.

Class Notation:	`<Text>Hello, World!</Text>`

Examples:	`Hello,%20World!`
	`Hello,\r\nWorld!`

Also See:	`Class, Font, FontStyle`

Line Breaks and Text Wrapping

If the `Height` is not specified, the text is automatically wrapped to multiple lines fit within the specified `Width`. Place `\r\n` in the text string to force a line-break at that location.

HTML Tags – *ImageGen Professional*

When using *ImageGen Professional*, you can include basic HTML tags in the `text=` parameter to change the style of the text at a character-by-character level. Unsupported tags are ignored. To display angle brackets, encode the < and > characters (see below).

`<b>`	Bold
`<i>`	Italic
`<em>`	Emphasis (italic)
`<strong>`	Strong emphasis (bold)
`<br />`	Line break

Multiple Fonts – *ImageGen Professional*

When using *ImageGen Professional*, you can change not only the font style for portions of generated text (with HTML Tags) but you can also change fonts for portions of the text with the `<font>` tag.

`<font name="blah">`	Change font

Any font installed on the server or defined and named in the `ImageGen.config` file can be used. Examples:

```
ImageGen.ashx?Text=abc<font name="Times New Roman">def</font>

ImageGen.ashx?Text=abc<font name="AvantGardeCE">def</font>
```

Multiple Fonts , Sizes, and Styles– *ImageGen Professional*

When using *ImageGen Professional*, you can easily change the font, font size, and font style for portions of the generated text.

<*classname*>	Change font, size, and/or style

In the `text=` querystring parameter, surround the portion of text with the name of a `<Class>` you have defined in the `ImageGen.config` file, within angle brackets (`< >`). For example, using the following <Class> definition in the ImageGen.config file…

```
<Class Name="LargeHeading">
    <Font>AvantGardeCE</Font>
    <FontSize>53</FontSize>
    <FontStyle>Italic</FontStyle>
</Class>
```

… you could simultaneously set the font, size, and style of a portion of the generated text as follows:

```
ImageGen.ashx?Text=abc<LargeHeading>def</LargeHeading>
```

Note: only the `<Font>`, `<FontSize>`, and `<FontStyle>` settings of a Class are applied to the text.

Text Encoding

Because *ImageGen* uses the querystring for the text= parameter, all text must be URL-encoded. URL-encoding is handled automatically and invisibly by browsers when you type directly into the location field of your browser. While it may appear that you typed a space between two words, the browser converted that space character into `%20`, the URL-encoded equivalent of a space. You will need to perform the URL-encoding yourself when using *ImageGen* within a website. You can encode text manually or with various website utilities (such as Umbraco's `UrlEncode()` function).

For *ImageGen Professional* users, you will also want to HTML-encode the text to display special characters, such as angle brackets (`< >`) and ampersands (`&`). Basically, you want to HTML-encode your text, and then URL-encode that string when using the `text=` parameter. Umbraco has both HTML and URL encoding functions in the Umbraco library so this can be entirely automated.

An Example

1. I want to print out 'a&b'
2. I then HTML-encode that text, which becomes 'a&amp;b'
3. I then URL-encode the HTML-encoded text, which becomes 'a%26amp;b'

HTML Entities

HTML entities (such as `&copy;`) are automatically handled appropriately by ImageGen Professional when you `urlencode(htmlencode(string))` them (becoming `%26amp;copy;`). HTML entities are not supported in *ImageGen Basic*, but you can use the equivalent numeric entities. For example, `&#169;` is equivalent to `&copy;`, which when encoded becomes `%26amp;%23169;` and is supported in both *ImageGen Basic* and *ImageGen Professional*.

TRANSPARENT

Make any color in the image that matches the `BgColor` transparent. Text rendered with `AntiAlias=True` determines the anti-aliasing based on the `FontColor` and `BgColor`. The `BgColor` is then made transparent. Though primarily useful for text graphics, images can also make use of the `Transparent` parameter. For GIF and PNG images only. Be aware that transparent GIFs may not retain their transparency when resized because of the potential for color shifting associated with the limited 8-bit GIF palette. Use PNG images with alpha transparency channels for best results when resizing.

Class Notation:	`<Transparent>True</Transparent>`
Options:	`True`, `False`
Default:	`True`

VALIGN

The vertical position of text in a rendered text-only image if `Height` is also specified. The vertical position of text or an overlay image atop an image, even if `Height` is not specified.

Class Notation:	`<VAlign>Middle</VAlign>`
Options:	`Top`, `Middle`, `Bottom`, `Near`, `Far`
Default:	`Middle`
Also See:	`Align`

VERSION

If `Version` is the only parameter specified, displays the version of *ImageGen* and all domains for which the Professional features are enabled. If any parameter other than `Version` is specified the `Version` parameter is ignored.

Example:	`ImageGen.ashx?version`
Also See:	`Class` (`HideDomains` property)

WIDTH

The width of the created or resized image. The `Image` will resize proportionally to the desired width unless `Height` is also specified. If both `Width` and `Height` are specified, the image is resized to those dimensions exactly. If `Constrain` or `Pad` is specified in addition to `Width` and `Height`, the image would be resized proportionally and padded to fill any extra space.

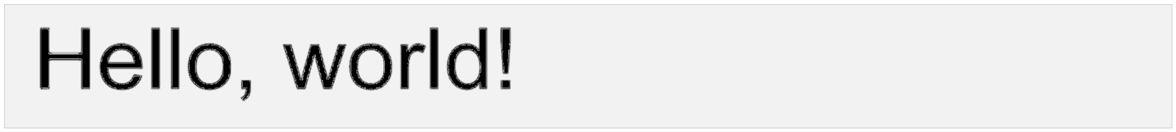
Class Notation:	`<Width>100 </Width>`
Default:	The width of the original image when resizing, or to the width required to contain the specified text
Also See:	`Constrain`, `Height`, `Pad`

Troubleshooting and FAQ

VERIFYING INSTALLATION

If no parameters are specified, *ImageGen* will display a simple "Hello, World!" image:

```
http://localhost/ImageGen.ashx
```



Hello, world!

CHECKING FOR PROFESSIONAL FEATURES

```
http://localhost/ImageGen.ashx?version
```

ImageGen will show the version installed, if the Basic or Professional features are enabled for the current domain, and all domains for which the Professional features are available. The output will be similar to the following:

```
ImageGen Professional version 2.9.x.xxxxx
Professional features are available for localhost
Professional features are available for *.local
Professional features are available for example.com
Professional features are available for sample.com
```

If the Professional features are not shown for your domains, ensure you have added the proper registration keys to the `/config/imagegen.config` file, and that the `web.config` file references the `imagegen.config` file. See the installation section for details.

Note that if the `/config/imagegen.config` file is not valid XML (tags don't open and close properly, for example) *ImageGen* will ignore the file entirely and operate in Basic mode rather than display a server error.

HIDING REGISTERED DOMAINS

Add `<HideDomains>true</HideDomains>` to the `ImageGen.config` file to disable listing the domains for which an *ImageGen Professional* key is installed when `imagegen.ashx?version` is requested. This will avoid advertising sites hosted on a server.

FILE NOT FOUND ERROR

If you receive a *File not found: image.png* error the requested `image` or `altImage` does not exist at the path specified. If you are using ImageGen Professional, double-check the `ImageBaseDir` setting in the `ImageGen.config` file as any `ImageBaseDir` setting is pre-pended to the `image` and `altImage` parameters.

NO TEXT DISPLAYED

If the `Text` and `FontSize` parameters generate text too large to fit within the specified `Width` and `Height`, no text will appear. If the `Height` is not specified, text will automatically wrap to fit the specified `Width`. If neither `Width` nor `Height` is specified the image will be as wide and tall as necessary to contain the text. Adjust the `Text`, `FontSize`, `Width`, and/or `Height` parameters to allow the text to fit and be displayed.

OUT OF MEMORY ERROR

An out of memory message typically means exactly that. It is generally caused by trying to resize massive original images with an application pool that doesn't have sufficient memory to accomplish the task. Either increase the memory available to the application pool or reduce the size of the original image.

This error also appears when using the `Text` parameter on Umbraco as a Service and Azure Websites (though not Azure Web Roles or VMs, nor traditional hosting) due to a limitation in those services. We are working on a solution. Until then, consider using overlay images or CSS for text atop images.

MEDIUM TRUST

ImageGen 2.x does not support Medium Trust environments.

USING IMAGEGEN IN LOAD-BALANCED ENVIRONMENTS

ImageGen works great in load balanced scenarios but each server must to manage its own 'cached' folders. Block the ‘cached’ folders from being synced or shared between servers.

CLIENT AND SERVER CACHING EXPLAINED

Server-side Caching

Server-side caching is a major feature of *ImageGen*. A request for a specific source image and its resizing parameters will create a file on the server’s drive to exactly those specifications. The next time the same image and parameters are requested, *ImageGen* will serve the image cached on the disk rather than wasting time, cpu power, and server memory to create it each time.

Further, if a website visitor has seen the resized image before, *ImageGen* will not bother to download the same image from its disk cache and waste time and bandwidth. Because *ImageGen* uses ETag response headers a small “304 Not Modified” response is sent allowing the browser to use the previously-downloaded image.

This technique allows *ImageGen* to immediately notice any changes to the original image (either local or remote) and generate and then cache updated files. It is very efficient and responsive.

Client-side Caching

With *ImageGen Professional*, there is also the possibility to cache images on the client’s browser and avoid even the request that would return a “304 Not Modified” response. Website performance is improved even more when images that won’t change frequently are cached for instant re-use on the client.

The `CachingTimeSpan` parameter element of a `<Class>` sets the number of seconds the website visitor’s browser should cache the image locally before issuing a request to determine if a new version of the image is available on the server.

Unless specified, there is no client-side caching, which is equivalent to `<CachingTimeSpan>0</CachingTimeSpan>`.

RESIZING IMAGES FROM REMOTE WEBSITES

Typically, *ImageGen* will resize images directly accessible from the website on the server's local drives. Sometimes you need to resize images hosted on a remote site or image server. To prevent your site from potentially being used as an image resizer for any image on the web, you need to specifically white list the domains that *ImageGen* is allowed to pull images from. Do be sure to respect copyright laws.

To white list a domain, update the `ImageGen.config` file with the domain. Please note that unlike registration key domains to activate *ImageGen Professional* features, white list domains must match precisely. For services such as *Flickr* that use multiple domains, you would list them all.

```
<RemoteDomainWhiteList>
   <Domain>farm1.static.flickr.com</Domain>
   <Domain>farm2.static.flickr.com</Domain>
   <Domain>farm3.static.flickr.com</Domain>
   <Domain>farm4.static.flickr.com</Domain>
   <Domain>farm5.static.flickr.com</Domain>
   <Domain>farm6.static.flickr.com</Domain>
</RemoteDomainWhiteList>
```

Or, use the * wildcard, which is particularly helpful with ever-changing CDNs:

```
<RemoteDomainWhiteList>
   <Domain>farm*.static.flickr.com</Domain>
</RemoteDomainWhiteList>
```

Using Remote Images

A local image is resized easily:

```
ImageGen.ashx?width=200&image=photo.jpg
```

A remote image is resized the same way, with the addition of the domain:

```
Imagegen.ashx?width=200&image=http://images.example.com/photo.jpg
```

You can hide the source of the remote image by creating a class and setting the `<ImageBaseDir>`:

```
<ImageGenConfiguration>
   …
   <Class Name="Remote">
       <ImageBaseDir>http://images.example.com/</ImageBaseDir>
       <Width>200</Width>
   </Class>
</ImageGenConfiguration>
```

With the above class, you would resize the remote image this way:

```
ImageGen.asxh?image=photo.jpg&class=remote
```

If the remote URL needs its own querystring parameters, be sure to URL-encode that URL's querystring. For example, if the remote URL to the photo were `http://images.example.com/images?q=ABCD&t=1`, URL-encode the querystring to become `http://images.example.com/images%3fq=ABCD%26t=1` (the `?` becomes `%3f` and the `&` becomes `%26` when URL-encoded).

With the example above, the ImageGen request would be:

```
/ImageGen.asxh?width=200&image=http://images.example.com/images%3fq=ABCD%26t=1
```

Default ImageGen.config File

```xml
<?xml version="1.0" encoding="utf-8" ?>
<ImageGenConfiguration>
   <Registration>
       <!-- ImageGen Professional is always available on localhost and *.local domains -->
       <Key domain="example.com">436DA8888A3BBC98662B23C6D1B635543E762854</Key>
       <Key domain="sample.com">55FD8FD97A199AB3082BC7DC11AE5F743B092C55</Key>
   </Registration>
   <RemoteDomainWhiteList>
       <Domain>www.sample.com</Domain>
   </RemoteDomainWhiteList>

   <!-- SAMPLE NAMED CLASSES, INCULDING THE SPECIAL 'DEFAULT' CLASS -->
   <Class Name="default" OverridesQueryString="true">
       <AllowUpsizing>true</AllowUpsizing>
       <MaxHeight>800</MaxHeight>
       <MaxWidth>800</MaxWidth>
   </Class>
   <Class Name="Thumbnail" OverridesQueryString="true">
       <InheritFromClass>default</InheritFromClass>
       <Width>200</Width>
       <Height>200</Height>
       <CachingTimeSpan>3600</CachingTimeSpan>
   </Class>
</ImageGenConfiguration>
```

Common Sense License

The key points of the license are:

- You can install *ImageGen* on as many computers as you wish.
- You agree not to decompile, reverse engineer or disassemble, modify or create derivative works.
- You agree not embed or otherwise include *ImageGen* in your applications without prior written agreement.

The full license details are:

Percipient Studios is the owner, developer and sole copyright holder of this product, which is licensed—not sold—to you on a non-exclusive basis. *ImageGen* is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties.

You may use the free version of *ImageGen* for as long as you like without ever having to acquire a registration key. However, it is recommended that you join the Percipient Studios mailing list so you can be kept informed of bug reports, usage tips and new releases as they become available.

The free version of *ImageGen* can be upgraded to enable additional features over and above those of the free product.

Each registration key covers a single second-level domain (e.g., example.com) and *ImageGen* can be installed on unlimited sites and machines within that domain. You need a unique registration key for each second-level domain for which you want the Professional features enabled.

By installing and/or using *ImageGen*, you agree NOT to:

(a) Decompile, reverse engineer or disassemble, modify or create derivative works based on *ImageGen* or the documentation in whole or in part.

(b) Remove any copyright or other Percipient Studios proprietary notices.

(c) Distribute any registration key for *ImageGen* to anyone other than the legally registered end user.

(d) Rent or lease *ImageGen* to any other party.

(e) Use a registration key that was not obtained directly from Percipient Studios.

You may transfer your *ImageGen* registration key to another person when transferring your domain only after receiving written authorization from Percipient Studios and only if the recipient agrees to be bound by the terms of this agreement.

Percipient Studios reserves the right to cancel the registration key(s) of any user who Percipient Studios determines is in violation of this agreement.

THE WARRANTIES IN THIS AGREEMENT REPLACE ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE IS PROVIDED "AS IS" AND PERCIPIENT STUDIOS DISCLAIMS AND EXCLUDES ALL OTHER WARRANTIES. IN NO EVENT WILL PERCIPIENT SUTDIOS BE LIABLE FOR ANY SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, EVEN IF WE HAVE KNOWLEDGE OF THE POTIENTIAL LOSS OR DAMAGE.

Version History

Version	Date	Key Points
2.9	Nov 2013	• Major performance improvements. • Enhanced cropping. • Fixed XSS vulnerability. • Errors and not-found results return 404 status. • Fixed file locking issue.
2.6 – 2.8	2012 - 2013	• Bug fixes for clients. • Not released.
2.5	Feb - Oct 2011	• Fixed XSS vulnerability. • Enhanced security. • Many new and improved features.
2.2	Sep 2008	• Added support for images hosted on remote servers. • Fixed caching problems, especially under high load.
2.1	Aug 2008	• Added support for multiple fonts and html tags within text • Not released.
2.0.2	Jun 2008	• Fixed error writing index.xml file when using FAT32 file-system.
2.0.1	Jun 2008	• Fixed bug with temporary xml files not being deleted.
2.0	May 2008	*ImageGen Basic* and *Professional* released. Many new and improved features, such as: • Added superimposing images on top of one another for watermarking and branding purposes, with alpha transparency support. • Added `ImageGen.config` classes to increase security and shorten URLs. • Maximum image size can be set. • Upsizing of images can be disabled. • Color images can be converted to grayscale output. • Rotate, crop, and flip images. • Text wrapping. • Improved error handling for busy sites.
1.6	Aug 2007	Added `AltBrowser` parameter.
1.5	Feb 2007	Fixed edge interpolation bug.
1.4	Dec 2006	Added `version` parameter. Added error handling in the event the `index.xml` file becomes corrupted.

1.3	Jul 2006	Added `constrain` parameter to image resizing.
1.2	May 2006	Minor bug fixes. Not released.
1.1	Jan 2006	Changed default location of cached text graphics to the `/data` folder. Fixed bug attempting to render apostrophe or ampersand in text.
1.0	Jan 2006	Initial release.