Image Layout

Images got a few properties in common with textframes and charts.

LineFormat and FillFormat

See the 'Shape-Layout' sample to see how to surround an image by a line or how to give it a background color.

Position

See the 'Shape-Layout' sample to see how an image receives its desired position.

Size

The size of an image can be determined in several manners.

Height and Width

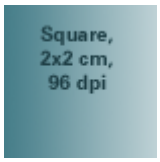
The size can be simply determined by defining both of the attributes **Height** and **Width**, as described in the 'Shape-Layout' sample.

Note that if both **Height** and **Width** are set, all other attributes described in this section will be ignored.

Default Size

The default size of an image is its size in pixels divided by its resolution. If an image file does not have resolution information, a default resolution of 72 dpi is taken to calculate its size. In our example, we got a portable network graphic of 76x76 pixels with a resolution of 96 dpi. This results in a rendered size of 2x2 cm.

```
\image("SquareWithColorGradient.png")
```

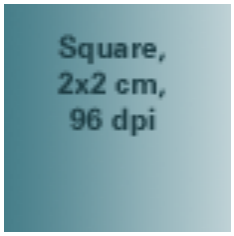


LockAspectRatio

If either **Height** or **Width** is given, the **LockAspectRatio** attribute determines the undefined attribute in each case.

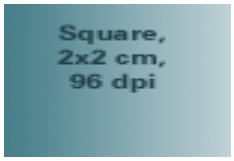
If **LockAspectRatio** is set to *true*, the image will be scaled equally in both horizontal and vertical direction to fit the given **Height** or **Width**, respectively. This is the default behavior.

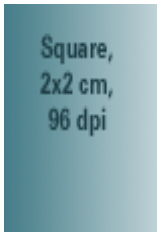
```
\image("SquareWithColorGradient.png")
[
  Width = "3cm"
  LockAspectRatio = true
]
```



If **LockAspectRatio** is set to *false*, the image will keep its original width or height, depending on which attribute is not set.

```
\image("SquareWithColorGradient.png")
[
  Width = "3cm"
  LockAspectRatio = false
  …
]
\image("SquareWithColorGradient.png")
[
  Height = "3cm"
  LockAspectRatio = false
  …
]
```

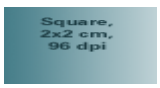


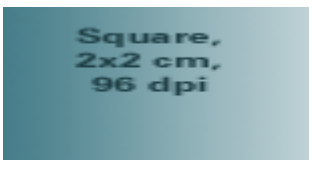


ScaleHeight and ScaleWidth

You can set vertical and horizontal scaling factors for the image.

```
\image("SquareWithColorGradient.png")
[
  ScaleHeight = 0.5
  LockAspectRatio = false
  …
]
\image("SquareWithColorGradient.png")
[
  ScaleWidth = 2.0
  LockAspectRatio = false
  …
]
```





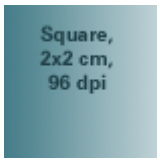
Note that you need to set **LockAspectRatio** to *false* here to receive the scaling in one dimension only. This is due to the default value of this attribute, which is *true*, as mentioned above.

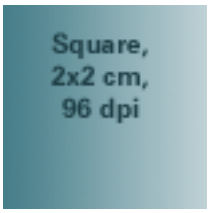
Resolution

Not all image files contain resolution information. Unfortunately, different programs then assume different default resolutions (72 or 96 dpi are the most common values).

To avoid trouble, you can specify a resolution that will be taken into account for size calculations.

```
\image("SquareWithColorGradient.png")
[
  Resolution = 96 //not necessary, because it's the picture's inherent resolution
  …
]
\image("SquareWithColorGradient.png")
[
  Resolution = 72
  …
]
```





PictureFormat

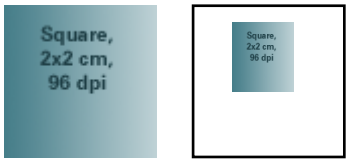
In the **PictureFormat** attribute you can find further attributes to crop your image.

In future versions of MigraDoc, further settings like a transparency key will be found here.

Cropping

An image can be cropped on each edge.

```
\textframe[Height = "2cm" Width = "2cm" …]
{
  \image("SquareWithColorGradient.png")
  [
    PictureFormat.CropLeft = "0.5cm"
    PictureFormat.CropTop = "0.2cm"
    PictureFormat.CropRight = "0.7cm"
    PictureFormat.CropBottom = "0.9cm"
    …
  ]
}
```



The uncropped image was put next to the cropped one to see the differences.

To visualize the cropping values, the cropped image is surrounded by a textframe of size 2x2 cm. Also for illustration, the cropped image is shifted downwards 0.2 cm, to the right 0.5 cm, corresponding to the values of **CropTop** and **CropLeft**.

Note that image cropping has no effect on HTML output.

Cropping and Image Size

It is important to mention that the values that are set for cropping are relative to the original picture size.

So if you change **Height** and **Width** in the previous sample, the image part being cut away will stay the same.

```
\image("SquareWithColorGradient.png")
[
  PictureFormat.CropLeft = "0.5cm"
  PictureFormat.CropTop = "0.2cm"
  PictureFormat.CropRight = "0.7cm"
  PictureFormat.CropBottom = "0.9cm"
  Height = "3cm"
  Width = "4cm"
  …
]
```

Here, the unscaled but cropped image from the previous sample was put next to the scaled one. Note that scaling is done so that the uncropped image would fit the given size.

Inline Images

In contrast to other shapes (textframes, charts), images can appear within paragraphs.

Overview

Inline images are handled like a word within a paragraph. This implies the following consequences:

- The bottom of an inline image is aligned to the text base line.
- The image height is taken into account when rendering the line containing it.
- All image attributes that deal with its position are ignored.
- The image position can be influenced by tab stops.

The following samples are typical applications for inline images.

Images as Bullets

Inline images are especially useful to create custom bullets.

```
\paragraph[Format.TabStops += {Position = "1cm"} …]
{
  \image("tick_green.png")[Height = "0.3cm"] \tab Point one is done
}
```

 Point one is done

Aligning with Tab Stops

An inline image can be aligned to a tab stop. This can be useful for logos in headers that shall appear right aligned.

```
\paragraph[Format.TabStops += {Alignment = Right Position = "16cm"} …]
{
  A Logo at a right aligned tab stop \tab
  \image("logo_e.gif")[Height = "1cm"]
}
```

A Logo at a right aligned tab stop