
ConTeXt basics for users: Images

Aditya Mahajan

Abstract

As the cliché goes, a picture is worth a thousand words. This article provides an overview of inserting pictures or images in a ConTeXt document.

A note about MkII and MkIV

In contrast to the previous articles in this series, from now on, I will assume that ConTeXt MkIV is being used: LuaTeX engine and PDF output. ConTeXt MkIV behaves differently from MkII, and in most cases provides additional features that are absent from MkII.

1 Basic usage

The simplest way to insert an image is to use:

```
\externalfigure[logo.pdf]
```



This command places the PDF image `logo.pdf` in a `\vbox`; the width and height of the image are equal to the natural dimensions of the image.

To set the width of the image to a specific size, say 1cm, use:

```
\externalfigure[logo.pdf][width=1cm]
```

Similarly, to set the height of the image to a specific size, say 2cm, use:

```
\externalfigure[logo.pdf][height=2cm]
```

If only the `width` or `height` of the image is specified, the other dimension is scaled appropriately to keep the aspect ratio.

To include a specific page, say page 5, of a multi-page PDF file, use:

```
\externalfigure[logo.pdf][page=5]
```

These four variations cover 90% of the use cases.

1.1 Natively supported file formats

ConTeXt natively supports the image formats enumerated below. The image format is determined from the file extension (case insensitive).

- PDF: File extension `.pdf`
- MPS (MetaPost output): File extension `.mps` or `.digits`
- JPEG: File extension `.jpg` or `.jpeg`
- PNG: File extension `.png`

- JPEG 2000: File extension `.jp2`
- JBIG or JBIG2: File extension `.jbig`, `.jbig2`, or `.jb2`

1.2 Including images after conversion

The image formats listed in Sec. 1.1 are the ones that may be embedded directly in a PDF. ConTeXt also supports a few other formats which are first converted to PDF using an external program. Of course, for such a conversion to work, the corresponding converter must be in the `PATH`.

Format	Extension	Converter
SVG	<code>.svg</code> , <code>.svgz</code>	<code>inkscape</code>
EPS	<code>.eps</code> , <code>.ai</code>	<code>gs</code> (or <code>gswin32c</code> on Windows): <code>Ghostscript</code>
GIF	<code>.gif</code>	<code>gm</code> from <code>GraphicsMagick</code>
TIFF	<code>.tiff</code>	<code>gm</code> from <code>GraphicsMagick</code>

The conversion generates a PDF file with prefix `m_k_i_v_` and a suffix `.pdf` added to the name of the original file. The result is cached, and the conversion is rerun if the timestamp of the original file is newer than that of the converted file.

1.3 Specifying image directories

By default, ConTeXt searches an image in the current directory, the parent directory, and the grand-parent directory.

To search for images in other directories, say a `./images` subdirectory and `/home/user/images`, use:

```
\setupexternalfigures
  [directory={images, /home/user/images}]
```

Note that one should always use forward slashes in path names, *even on Windows*.

The default search order is: the current directory, the parent directory, the grand-parent directory, and then the paths specified by the `directory` key. To restrict image search only to the paths specified by the `directory` key, use:

```
\setupexternalfigures
  [location=global]
```

To restore the default search behavior, use:

```
\setupexternalfigures
  [location={local,global}]
```

The ConTeXt distribution includes three sample images: `cow.pdf`, `mill.png`, and `hacker.jpg`, that are useful when creating minimum working examples to illustrate a bug on the mailing list. These images are located in the `TEXMF` directory. To add the `TEXMF` directory to the image search path, use:

```
\setupexternalfigures
  [location={local,global,default}]
```

The above alternative adds the *entire* TEXMF directory to the search path, *including the doc/ directory!* Therefore, one needs to be extremely careful when using this option. In fact, I would advise not using `location=default` except for illustrative minimal working examples.

1.4 Including remote images

Like all other ConTeXt macros that read files, `\externalfigure` also supports reading remote files from HTTP(S) servers. An example:

```
\externalfigure
  [http://tug.org/images/logobw.jpg]
```



When a document containing a remote file is compiled for the first time, the remote file is downloaded from the server and stored in the LuaTeX cache directory. This cached file is used during subsequent runs.

Normally, the remote image is downloaded again if the image in the cache is older than 1 day. To change this threshold to, for example, 2 minutes (120 seconds), either add `\enabledirectives[schemes.threshold=120]` in the ConTeXt file, or compile the ConTeXt file using the command

```
context --directives=schemes.threshold=120 \
  filename
```

The variable `schemes.threshold` is global, so changing its value affects all other macros like `\input`, `\usemodule`, `\component`, etc. that load remote files.

2 Image transformations

2.1 Scaling images

To scale an image use the `scale` key: `scale=1000` corresponds to the original dimensions of the image, `scale=500` scales the image to 50% of the original size, `scale=1500` scales the images to 150% of the original size, and so on. For example:

```
\externalfigure[logo.pdf][scale=500]
```



Use `\setupexternalfigures` to set the scale of all images. For example, to scale all images to be twice their original size, use:

```
\setupexternalfigures[scale=2000]
```

If either `width` or `height` is specified, then the `scale` key has no effect.

In addition, the `xscale` and `yscale` keys scale the image in only one dimension. For example:



`xscale=500`



`yscale=500`

2.2 Specifying maximum size of an image

Often, we want the included image to be no larger than a given size. E.g., this ensures that an included image is no more than `0.2\textwidth`:

```
\externalfigure[logo.pdf]
  [maxwidth=0.2\textwidth]
```



If `maxwidth` is specified and the width of the image is less than `maxwidth`, then the image is not scaled; if the width of the image is greater than `maxwidth`, then the width is restricted to `maxwidth` and the height is scaled appropriately to maintain the original aspect ratio.

Analogous to `maxwidth` is the option `maxheight`, which checks the height of the image.

In my own style files, I usually specify the following to ensure that figures do not overflow the text area:

```
\setupexternalfigures
  [maxwidth=\textwidth,
  maxheight=0.8\textheight]
```

2.3 Rotating images

To rotate the included image by 90, 180, or 270 degrees, use the `orientation` key. For example:

```
\externalfigure[logo.pdf][orientation=90]
```



To rotate by an arbitrary angle, use the `\rotate` command. For example:

```
\rotate[rotation=45]
  {\externalfigure[logo.pdf]}
```



2.4 Mirroring images

To mirror (flip) an image, use the generic `\mirror` command. For example, to mirror horizontally:

```
\mirror{\externalfigure[logo.pdf]}
```



To mirror vertically, first rotate the image by 180° and then mirror it:

```
\mirror{\externalfigure[logo.pdf]
         [orientation=180]}
```



2.5 Clipping images

To clip an image, use the generic `\clip` command. For example, to clip the original image to a 1cm x 2cm rectangle at an offset of (3mm, 5mm) from the top left corner:

```
\clip[width=1cm, height=2cm,
      hoffset=3mm, voffset=5mm]
      {\externalfigure[logo.pdf]}
```



As another example, this cuts the image into a 3x3 pieces and then outputs the (2,2) piece:

```
\clip[nx=3,ny=3,x=2,y=2]
      {\externalfigure[logo.pdf]}
```



3 Troubleshooting

3.1 Visualizing the image bounding box

If, for instance, the image is taking more space than expected, it can be useful to visualize the bounding box of the image. To do this:

```
\externalfigure[logo.pdf] [frame=on]
```



ConTeXt includes a Perl script `pdftrimwhite` that removes extra white space at the borders of a PDF file. To run this script:

```
mtxrun --script pdftrimwhite [flags] input output
```

The most important flag is `--offset=<dimen>`, which keeps some extra space around the trimmed image.

Similar functionality is provided by another Perl script, `pdfcrop`, that is included in most TeX distributions.

3.2 Tracking what is happening

To get diagnostic information about image inclusion, enable the tracker `graphics.locating` either by adding

```
\enabletrackers[graphics.locating]
```

in the ConTeXt file, or by compiling the ConTeXt file using the command

```
context --trackers=graphics.locating filename
```

The tracker writes diagnostics to the console. Suppose we use `\externalfigure[somefile.pdf]` and ConTeXt finds the file in the current search path; then the following information is printed on the console: (The tracking messages here are formatted to typeset nicely. The actual messages are slightly different.)

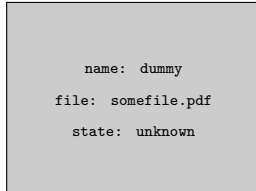
```
graphics > inclusion > locations: local,global
graphics > inclusion > path list: . . . /..
graphics > inclusion > strategy: forced format pdf
graphics > inclusion > found: somefile.pdf ->
                                somefile.pdf
graphics > inclusion > format natively supported
                                by backend: pdf
```

If the file `somefile.pdf` is not found in the current search path, then the following information is printed on the console (even if the `graphics.locating` tracker is not set):

```
graphics > inclusion > strategy: forced format pdf
graphics > inclusion > not found: somefile.pdf
graphics > inclusion > not found: ./somefile.pdf
```

```
graphics > inclusion > not found: ../somefile.pdf
graphics > inclusion > not found: ../../somefile.pdf
graphics > inclusion > not found: images/somefile.pdf
graphics > inclusion > not found:
/home/user/images/somefile.pdf
graphics > inclusion > format not supported:
```

and a placeholder gray box is put in the output:



Sometimes, one would rather use a placeholder image for an image that is yet to be made. In such cases, load the MP library `dum` via:

```
\useMPlibrary[dum]
```

Then, whenever an image file is not found in the current search path, a random MetaPost image is shown in the output.



3.3 Images at the beginning of a paragraph

Using `\externalfigure[...]` at the beginning of a paragraph results in a line break after the image. This is because `\externalfigure` is a `\vbox` and when \TeX encounters a `\vbox` at (what appears to be) the beginning of a paragraph, it remains in vertical mode. To prevent this, add `\dontleavehmode` before `\externalfigure`, like this:

```
\dontleavehmode
\externalfigure[...] ... first line ...
```

4 Settings for multiple images

4.1 Image settings

Suppose your document contains many side-by-side images, and you want all of these images to be of the same size. In addition, you want to control the size of all images by changing only one setup. To do this, you can use the `\defineexternalfigure` macro, which defines a named collection of image

settings. For example, to define a collection where the image width is 3cm, use:

```
\defineexternalfigure[logo-settings]
[width=3cm]
```

And then to use these settings in an image, use:

```
\externalfigure[group.pdf][logo-settings]
```



or, if you want to add or override settings, use:

```
\externalfigure[group.pdf][logo-settings]
[height=2cm]
```

4.2 Labeled images

Suppose your document contains an image at multiple locations; all of these images are to be of the same size, which is not necessarily the same as the natural size of the image. Furthermore, as before, you want to set the size of all the images by changing only one setup. Here, the macro to use is `\useexternalfigure`, which defines a symbolic label for inserting an image plus settings. For example:

```
\useexternalfigure[mylogo]
[logo.pdf][width=2cm]
```

defines an image label `mylogo` that maps to the image file `logo.pdf` and sets its width to 2cm. This image label may be used as a normal image filename:

```
\externalfigure[mylogo]
```



5 Conclusion

In this article, I briefly explained how to include images in your document. Usually, one wants the included images to have a number and a caption — i.e., to display the image in a float. I will discuss floats in a future article in this series.

◇ Aditya Mahajan
adityam (at) ieee dot org