# A Calligraphy Grid in ConTeXt/LMTX

**WARNING: WORK IN PROGRESS! IT MAY NOT WORK! DOCUMENTATION MAY BE INCOMPLETE OR INCORRECT! USE AT YOUR OWN RISK!**
This project is a generator of grids for medieval calligraphy in ConTeXt/LMTX and MetaPost.

## Files included

- Main module: `tex/context/third/pauta/t-pauta.mkxl`
- Example file: `doc/context/third/pauta/pauta-example.tex` (also in pdf)
- A copy of this file in ConTeXt format: `doc/context/third/pauta/pauta-example.tex` (also in pdf)
- Environment file for the examples (not needed for Pauta to work): `doc/context/third/pauta/env-pauta.tex`
- This `README.md` file, and a pandoc generated ConTeXt version at `doc/context/third/pauta/README.tex`. It gets regenerated by the `build.lua` script on each run using pandoc.
- The build script `build.lua`, that takes care of generating the tex version of the README file and all the pdfs.

## Use

1. Clone this repository: `git clone https://github.com/conradolandia/pauta.git`, and `cd` inside it.
2. Copy the `doc` and `tex` folders to your ConTeXt tree and rebuild your database with `context --generate`. You can find more details about the process on the ConTeXt wiki. Alternatively, call context with the `--path` flag, and provide it with the path of this folder, i.e: `context --path=/home/user/pauta`. Alternatively still, simply place `t-pauta.mkxl` on the same directory as the file importing it.
3. Invoke the `\Pauta` macro as many times as you want pages. Each invocation can have a different configuration. Each invocation will create one single page.
4. The data about the hand[1] is autogenerated by the module and set into the header or the footer, following the user configuration.
5. **WARNING:** This module takes over the top and bottom areas, and does not reset them properly yet. So if your document includes other content on those areas you will need to reset again to your liking by manually invoking `\setuptoptexts`/`\setupbottomtexts`. This will be hopefully improved in the future.

## Generating the example file

Review and run the lua script `build.lua`. If you don't have a standalone lua interpreter, you can run it with luametatex like so:

```
luametatex --luaonly build.lua
```

[1] This a way of saying the *font* in fancy calligraphic terms.

You can also adapt it and use to build your own projects. `build.lua` also lives in its own repository.

# Configuration Parameters

All parameters are optional. Defaults are as follows:

```
\Pauta[
  hand=, % Hand name. If not defined, will not show info on the right of the header / footer
  handInfo=, % Some extra info for the hand. If not defined, will not show info on the right of the
header / footer
  infoPosition=header, % Where to show the extra info (header | footer)
  infoLeft={\setup{pauta:content:leftmark}}, % If defined, will override autogenerated hand info on
the left of the footer / header
  infoRight={\setup{pauta:content:rightmark}}, % If defined, will override autogenerated hand info
on the right of the footer / header
  displayNibs=false, % Show nib-width marks (true | false)
  displayAngleMarks=false, % Display dotted guides for the nib angle (true | false)
  nibWidth=3mm, % Pen nib width (must include units, or it will default to big points)
  nibAngle=35, % Nib working angle in degrees
  ascenders=3, % Number of ascender lines (in nib widths)
  xHeight=4, % Number of x-height lines (in nib widths)
  descenders=3, % Number of descending lines (in nib widths)
  adjustment=0, % Sometimes it's necessary to adjust the line height if is longer than TextHeight,
still not sure why it happens but it happpens... a value of 1 or 2 should solve it.
  mainColor={s=.4}, % Main color (lines that separate sections)
  secondaryColor={s=.6}, % Secondary color (lines separated by a nib width and dotted angle lines)
  tertiaryColor={s=.8}, % Tertiary color (nib width marks on the left margin)
]
```

# Code Examples

## Example 1: Basic Usage

```
\usemodule[pauta]

\startdocument
\Pauta[
  hand={Carolingian},
  handInfo={Tours school, VIII\high{th} century},
  infoPosition=header,
  displayNibs=true,
  displayAngleMarks=true,
  nibWidth=3mm,
  ascenders=2,
  xHeight=3,
  descenders=2,
  adjustment=0,
  mainColor={s=.6},
  secondaryColor={s=.8},
  tertiaryColor={s=.8},
]
```

```
\stopdocument
```

## Example 2: Multiple Pauta Instances

```
\usemodule[pauta]

\startdocument
\Pauta[
 hand={Carolingian},
 handInfo={Tours school, VIII\high{th} century},
 infoPosition=header,
 displayNibs=true,
 displayAngleMarks=true,
 nibWidth=3mm,
 ascenders=2,
 xHeight=3,
 descenders=2,
 adjustment=0,
 mainColor={s=.5},
 secondaryColor={s=.6},
 tertiaryColor={s=.7},
]

% Overriding the hader / footer info:

\Pauta[
 infoLeft={An excercise in Visigothic script},
 infoRight={from an Spanish manuscript, VII\high{th} century},
 infoPosition=footer,
 displayNibs=true,
 displayAngleMarks=false,
 nibWidth=2mm,
 ascenders=4,
 xHeight=3,
 descenders=4,
 adjustment=1,
 mainColor={s=.3},
 secondaryColor={s=.4},
 tertiaryColor={s=.5},
]
\stopdocument
```

This project aims to provide a somewhat flexible and kind of efficient tool for creating calligraphy practice templates, leveraging the power of ConTeXt and MetaPost.

# Changelog

- 20240307: Added build script, first alpha version
- 20240308: Improved build script, TDS compliant branch created
- 20240311: Improved build script yet even more, and we are TDS compliant.