Installing ConTEXt

The organization of files

There are several ways to install ConTEXt LMTX. Download the archive(s) and compile the binaries is one. But normally one also needs fonts and maybe other resources. A ConTEXt installation is normally organized like this:

```
data/context/tex/texmf
data/context/tex/texmf-context
data/context/tex/texmf-<architecture>
data/context/tex/texmf-fonts
data/context/tex/texmf-project
```

The second tree can be part of the first tree, but updates are distributed as archives that unpack in a tree which normally then is the `texmf-context` tree. Examples or architectures are:

```
data/context/tex/texmf-win64
data/context/tex/texmf-linux-64
data/context/tex/texmf-linuxmusl-64
data/context/tex/texmf-linux-aarch64
data/context/tex/texmf-osx-64
data/context/tex/texmf-osx-arm64
data/context/tex/texmf-freebsd-amd64
data/context/tex/texmf-openbsd-amd64
```

In the architecture tree we find the binaries:

```
data/context/tex/texmf-win64/bin/luametatex
data/context/tex/texmf-win64/bin/context       -> luametatex
data/context/tex/texmf-win64/bin/mtxrun        -> luametatex
data/context/tex/texmf-win64/bin/context.lua
data/context/tex/texmf-win64/bin/mtxrun.lua
```

Those are all that you need. Of course you can put more in there but because ConTEXt ships with extra scripts (located elsewhere) nothing more is needed for regular edit-run-preview cycles.

Installation

We have a rather minimalistic installation that can be easily updated too. The installer works mostly the same for all platforms. It starts with fetching an installation zip, like:

- lmtx.pragma-ade.nl/install-lmtx/context-win64.zip
- lmtx.pragma-ade.nl/install-lmtx/context-linux-64.zip
- lmtx.pragma-ade.nl/install-lmtx/context-linuxmusl-64.zip
- lmtx.pragma-ade.nl/install-lmtx/context-linux-aarch64.zip
- lmtx.pragma-ade.nl/install-lmtx/context-osx-64.zip
- lmtx.pragma-ade.nl/install-lmtx/context-osx-arm64.zip

- lmtx.pragma-ade.nl/install-lmtx/context-freebsd-amd64.zip
- lmtx.pragma-ade.nl/install-lmtx/context-openbsd-amd64.zip

If that doesn't work try without the `install-lmtx` because we might adapt this depending on experiments (like we have an `install-beta` every now and then).

You unpack this archive in an accessible location. On production servers we use `/data/context` (which can be a mount on a fileserver). On a personal machine it can be the home directory. But in any case it makes sense to use a simple path without spaces in the name. On MS Windows `c:/data/context` is a good place.

When you have unzipped the archive the process is different for MS Windows and Unix. On MS Windows, you can open a console and run the install command, but easier is to just select the `install.bat` file and run it (right mouse button menu), preferable as administrator, because that will use symbolic links instead of copies of files, which saves space. There is also a `setpath` script that will add the binary path to the global path.

On Unix you execute the `install.sh` script from a console. You might need to change its permission. You might want to add the binary path to your regular path.

At the top of the `install.[bat|sh]` script you will find a variable that defines the list of servers. You can adapt that list to your needs.

Updating

An update is simple: just run the install script again. When you do a fresh installation, zipfiles will be downloaded and unpacked, but an update will just fetch changed files.

The website where you fetch the files can change. In the install file you can adapt the address if that happens.

When you decide to do a new installation, you can just download the installation zip, unzip it, remove the `texmf` and `texmf-context` paths but keep `texmf-project` and `texmf-font` as that is where you put your personal files.

Running

You can add the `/tex/texmf-.../bin` path to your regular path variable but you can also run the `context` and `mtxrun` commands with a full path. If you run from an editor that is normally easy to configure. You need of course a pdf viewer too. A simple test file is:

```
\setupbodyfont[pagella]
\starttext
    \input tufte
\stoptext
```

Say that you save this as `test.tex` someplace, then you process it with:

```
context test.tex
```

The first time fonts are used they will be converted to a format suitable for LuaMetaTEX, but successive runs will used the cached copy.

You can also run LuaTEX but for that you need to generate the database with `context`:

```
context --luatex --generate
context --luatex --make
```

This then permits:

```
context --luatex test.tex
```

Support

You can get support at:

maillist	`ntg-context@ntg.nl / http://www.ntg.nl/mailman/listinfo/ntg-context`
webpage	`http://www.pragma-ade.nl / http://context.aanhet.net`
archive	`https://github.com/contextgarden`
wiki	`http://contextgarden.net`