Filenames

The ConTEXt distribution follows a rather strict organisation. Originally all files that implemented macros had names using the pattern:

`xxxx-xxx.tex`	TEX file
`mp-xxxx.mp`	MetaPost file

You can still find files that conform to these patterns but the organization evolved. The `tex` suffix is now normally used just for documents.

Styles and modules can have names of any length, and can be recognized by their prefix:

`s-aaaa.tex`	style (rendering related)
`m-aaaaaa.tex`	module (functionality related)
`x-aaaaa.tex`	xml module (functionality related)

ConTEXt MkII, the frozen version for pdfTEX and XETEX, uses names like:

`context.mkii`	the main TEX file
`cont-xx.mkii`	an interface specific TEX file
`xxxx-xxx.mkii`	TEX file
`mp-xxxx.mpii`	MetaPost file
`metafun.mpii`	the main MetaFun file

ConTEXt MkIV, the current version, has files with names like:

`context.mkiv`	the main TEX file
`cont-xx.mkiv`	an interface specific TEX file
`xxxx-xxx.mkiv`	TEX file
`xxxx-xxx.mkvi`	TEX file with named parameters
`xxxx-xxx.mkix`	file with lmx template
`xxxx-xxx.mkxi`	file with lmx template with named parameters
`xxxx-xxx.lua`	a file with Lua code
`xxxx-xxx.lfg`	so called font goodie Lua files
`metafun.mpiv`	the main MetaFun file
`minifun.mpiv`	a subset of MetaFun
`mp-xxxx.mpiv`	MetaPost file

There are more suffixes used, like `tua` and `tuc` for multipass jobdata, and `log` for log files. In the cache tree you can run into `luv`, `lui`, `luj` and `lum` for startup data, `tma` for Lua cache files, `tmb` for LuajitTEX bytecode, `tmc` for LuaTEXbytecode and `tmd` for LuaMetaTEX bytecode, but you can forget about them.

There can be files with `-imp-` in the name: these relate to other files with a similar name.

The follow up on MkIV is called lmtx (or MkXL?) and is compatible with MkIV: it uses, at least now, mostly the same code. But, as it depends on LuaMetaTEX it also has some different internals. Therefore you will find some additional files:

`context.mkxl`	the main TEX file
`cont-xx.mkxl`	an interface specific TEX file
`xxxx-xxx.mkxl`	TEX file
`xxxx-xxx.mklx`	TEX file with named parameters

metafun.mpxl	the main MetaFun file
minifun.mpxl	a subset of MetaFun
mp-xxxx.mpxl	MetaPost file

This means that a file xxxx-xxx can be present with any of the mk.. suffixes. In the standard distribution the MkII and MkIV files have their own path (directory), and lmtx only ships what it needs.

This somewhat complicated setup is needed in order to support both LuaTEX and LuaMetaTEX system. The more LuaMetaTEX diverges from LuaTEX, the more the codebase will be split so eventually we might end up with MkII, MkIV and lmtx as more or less independent versions. We try to share the Lua code as much as possible, also because some components are generic.

The name pattern xxxx- groups the files in categories. These are also referred to from the interface definitions. Examples of categories are syst for system modules that define various low level support macros and mechanisms. The user interface is handled by modules in the mult namespace. The supp modules layer on top of that and provide more helpers. The font and type modules deal with fonts, lang handles language support. The strc modules implement structural components, tabl does tables and page handles the layout. Specialized categories like mlib, meta and grph are for graphics, and publ is used for the publication (bibliography) subsystem. Just to give you an idea.