LuaTEX

using plain TEX

TUG 2009
Although we use ConTeXt for testing we also need to check basic behaviour with a minimal macro set and bare definitions. Most tests with plain TEX are done with no additional code loaded (like running the TExbook). It can be handy to check at least basic OpenType font support in plain TEX. And using mplib in its simple form also makes sense.
We don't touch the plain TEX format. Format generation happens via `luatex-plain.tex`. There some code is hooked into `\everyjob` so that a few extra files are loaded. Additional font support uses files from the ConT EXt distribution but fancy features are not available. Also MetaPost library support is loaded, but again with no fancy extensions. In order not to get burdened by ConT EXt beta versions, all font related code can be loaded from a merge file.
As there is no high level font interface, the input method is similar to \TeX but we also support location prefixes (\texttt{file:} and \texttt{name:}) and have a few more keys.

Fonts can be loaded in base mode in which case \TeX's traditional mechanisms for ligature building and kerning are used.

We can also use node mode where we use Lua instead. This is needed when we use more complex OpenType features.

We do support OpenType math fonts but keep in mind that \TeX{} has its own font model and math setup as it's not an all-purpose macro package.
What it does

The extensions are loaded as follows:

\input plain
\directlua {tex.enableprimitives('', tex.extraprimitives())}
\pdfoutput=1
\everyjob \expandafter{\the\everyjob\input luatex-basics\relax\input luatex-fonts\relax\input luatex-mplib\relax}
\edef\fmtversion{\fmtversion+luatex}

The basics provide a newattribute macro. This is needed because we use a reserved range at the Lua end.
You need to make a format:

```
luatex --ini luatex-plain.tex
```

The format has to be copied to a known path:

```
copy luatex-plain.fmt <texroot>/texmf-mswin/web2x/luatex/luatex.fmt
```

After that you can process files:

```
luatex yourfile.tex
```

Don’t forget:

```
\pdfoutput=1
```
Defining fonts

If you want access by name you need to generate a font database, using:

```latex
tmtxrun --script font --names
```

and put the resulting file in a spot where LuaTEX can find it.
Running MetaPost

beginfig(1) ;
draw fullcircle scaled 10cm withcolor red withpen pencircle xscaled 4mm yscaled 2mm rotated 30 ;
endfig ;
As a bonus, the ConTEXt distribution ships with a Font Inspector.