

ConT_EXt und LMTX – (wozu) will man das?

Leah Neukirchen

<leah@vuxu.org>

BayT_EX 2023, Kirchheim bei München

28. Juli 2023

ConT_EXt im Kontext der Zeit

ConT_EXt ist ein T_EX-Makropaket, das seit 1990 von Hans Hagen für firmeninterne Zwecke geschrieben wurde.

Um 1995 wurde eine Version namens „Mark II“ veröffentlicht. Diese hatte dann Unterstützung für PDF.

Mit der Entwicklung von LuaT_EX entstand 2005 die Version „Mark IV“, die intern vieles in Lua realisiert.

Ab 2019 wurde eine neue Engine LuaMetaT_EX programmiert und 2022 als Open Source veröffentlicht. Diese stellt den aktuellen Entwicklungszweig da.

Wodurch zeichnet sich ConT_EXt aus?

Im Vergleich zu L^AT_EX ist ConT_EXt monolithisch:

(Fast) Alle Features stammen aus einer Hand und passen oft gut zusammen.

Es werden nur einige wenige „third-party“-Module angeboten (z. B. TikZ).

Nachteil: nicht jedes obskure Makropaket von CTAN geht unter ConT_EXt.

Erwähnenswerte spezielle Features: registerhaltiger Satz, Bogenmontage, Einlesen von XML.

Wodurch zeichnet sich ConT_EXt aus?

Die Grundfunktionalität ist gut organisiert und sehr flexibel konfigurierbar. (Man denke an KOMA-Script statt L^AT_EX.)

Prinzip: ConT_EXt erlaubt Trennung von (semantischem) Markup und Aussehen; es ist einfach, neues Markup zu definieren.

ConT_EXt hat nur wenig Markup vordefiniert (z. B. Überschriften), der User definiert selbst, was gebraucht wird und wie es aussehen soll.

Konzepte

Instantiierung und Parametrisierung mit Key/Values:

Für ein **xyz** (z. B. `head`, `enumeration`, `note`) gibt es:

- `\definexyz[abc]` um ein neues **xyz** zu erzeugen.
- `\setupxyz[abc][foo=bar, ...]` um das **abc** zu konfigurieren.
- `\startabc[...]`
...
`\stopabc` um das **abc** zu verwenden.

Demo I

```
\definesymbol[todo][\raisebox{1pt}\hbox{\getglyph{DejaVuSans}{\char"25CB}}]  
\definesymbol[done][\raisebox{1pt}\hbox{\getglyph{DejaVuSans}{\char"25C9}}]
```

```
\defineitemgroup[todo]  
\setupitemgroup[todo]  
  [symbol=todo, color=darkred]  
\setupitemgroup[todo][packed]
```

```
\starttexdefinition itemdone  
\sym{\symbol{done}}  
\stoptexdefinition
```

Einkaufsliste:

```
\starttodo  
\item Milch  
\itemdone Brot  
\itemdone Salz  
\stoptodo
```

Einkaufsliste:

- Milch
- Brot
- Salz

Demo II

```
\defineoverlay
  [dbend]
  [{\framed[frame=off,width=\overlaywidth,height=\overlayheight,align=top]{
    \offset[x=0.666cm,y=0.666cm]{\scale[width=1cm]{\symbol{dbend}}}}}]

\definecolor[bg:red][h=1,s=0.2,v=1]
\defineframedtext
  [danger]
  [width=max,loffset=2cm,
   background={color,dbend},
   backgroundcolor=bg:red,
   framecolor=red,
   rulethickness=3pt,align=flushleft]

\startdanger
{\bf Privatgrundstück} \par
Betreten auf eigene Gefahr. Eltern
haften für ihre Kinder.
\stopdanger
```



Privatgrundstück

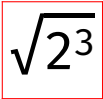
Betreten auf eigene Gefahr.
Eltern haften für ihre Kinder.

Lua

Quasi alle Funktionalität von ConT_EXt ist auch über eine Lua-API ansteuerbar:

```
\startluacode
context.startsection {title="A section",label="first"}
context.framed({ framecolor = "red" },
  function() context.mathematics("\sqrt{2^3}") end)
\stopluacode
```

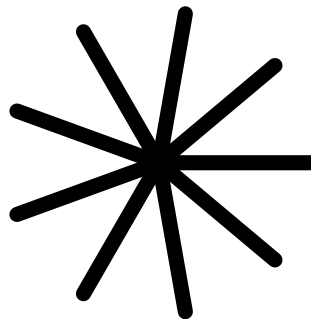
1 A section


$$\sqrt{2^3}$$

luamplib

Durch die enge Integration von ConT_EXt und MetaPost dank mp_lib können grafische Elemente flexibel und schnell eingesetzt werden.

```
\startMPcode
def star(expr size, n) =
  for a=0 step 360/n until
    round(360*(1-1/n)):
    draw (origin -- (size/2,0))
      rotatedaround (origin,a);
  endfor;
enddef;
```



```
pickup pencircle scaled 2mm;
star(4cm,9);
\stopMPcode
```

Wie komme ich an ConT_EXt?

Es gibt einen Binärinstaller für LMTX für Linux, MacOS, Windows, FreeBSD und OpenBSD. Selbst kompilieren ist auch nicht schwierig.

Vorteil: das System braucht nur knapp 250 MB (Fonts und Doku sind dabei).

Ab T_EXLive 2023 ist LMTX auch dabei, wird aber nicht regelmäßig aktualisiert.

Über LMTX

LuaMetaT_EX ist eine Weiterentwicklung von LuaT_EX, die radikal entschlackt wurde. Sämtlicher Quellcode wurde nach C konvertiert und ist direkt kompilierbar.

Es gibt keine eingebauten Backends mehr, der PDF-Output wird rein in Lua realisiert. Viele andere Teile werden auch in Lua implementiert.

Derzeit ist LMTX nur mit ConT_EXt benutzbar (ein inoffizielles Projekt versucht L^AT_EX zu portieren).

Features von LMTX

Vorteile: gute Performanz, viele technische Feinheiten und noch viel mehr Möglichkeiten in der Zukunft.

- Silbentrennung, Ligaturen und Kerning finden getrennt statt.
- Unterstützung für „OpenType Math“-Schriften.
- SVG-Rendering mit MetaPost.

Ein Geschwindigkeitsvergleich

Mein Romanprojekt (672 Seiten, fast nur Fließtext ohne komplexe Logik):

Engine	Version	Zeit
LuaT _E X 1.17.0	MKIV 2023.05.05	39 s
LuaMetaT _E X 2.10.09	MKXL 2023.06.22	15 s

Ressourcen

- Die Mailingliste `ntg-context`
- Die Website `https://contextgarden.net`
- *A not so short introduction to ConT_EXt* von Joaquín Ataz-López
- *The definitive guide to ConT_EXt MkIV and LMTX documentation*, gesammelt von Henri Menke
- *Drucksachen gestalten mit ConT_EXt und LuaMetaT_EX* von Henning Hraban Ramm (im Erscheinen)

Gesamturteil

ConT_EXt eignet sich gut wenn man:

- eigenständige, komplexe Dokumente selbst gestalten will
- bereit ist, einige Zeit auf Lernen und Ausprobieren aufzuwenden (ConT_EXt verwenden ist relativ einfach, komplexe Setups erstellen nicht.)
- die Flexibilität von XML-Input oder Lua benötigt
- ein kompaktes und schnelles T_EX-System wünscht

Gesamturteil

ConT_EXt eignet sich *nicht* wenn man:

- eine vorgegebene L^AT_EX-Formatvorlage verwenden muss
- sehr spezieller L^AT_EX-Makropakete bedarf
- vollständige und ausführliche Dokumentation benötigt
- in 30 Jahren den gleichen Output wie heute erwartet (LMTX ist allerdings archivierbar)

Trotzdem gilt: Viele ConT_EXt-Features haben wir auch in L^AT_EX dank LuaT_EX.

Vielen Dank!

Noch Fragen?