cweb Source Code for T_E_X
Why, How, and What

Martin Ruckert, August 2021

85. Here the author of T_{\text{e}}X apologizes for making use of the numerical relation between 'Q', 'R', 'S', and the desired interaction settings batch.mode, nonstop.mode, scroll.mode.

\begin{verbatim}
(Change the interaction level and \texttt{return} 85) =
{
   error.count = 0;
   interaction = batch.mode + c - 'Q';
   print("OK, \text{entering}");
   switch (c) {
   case 'Q':
      { print.esc("batchmode");
       decr(selector);
      } break;
   case 'R': print.esc("nonstopmode"); break;
   case 'S': print.esc("scrollmode");
   } /* there are no other cases */
   print("...");
   print ln();
   update_terminal;
   return;
}
\end{verbatim}

This code is used in section 83.
Why?

Why do we want cweb Source Code of \TeX{}?

- Toolchain
- Modification
- Debugging
- Education
- Experiments

Example: ktex.w

- tex.web
  + etex.ch
  + 32bit pointer type
  + kpathsea library
  = ktex.w

- It is not the purest implementation of \TeX{}.
- It is a useful basis.
- It is still close to the original \TeX{}. 
Why do we want cweb Source Code of \TeX\? 

- **Toolchain**
- Modification
- Debugging
- Education
- Experiments
Why?

Why do we want cweb Source Code of $\TeX$?

- Toolchain
- Modification
- Debugging
- Education
- Experiments
Why?

Why do we want cweb Source Code of \( \text{T}_{\text{E}}\text{X} \) ?

• Toolchain
• Modification
• Debugging
• Education
• Experiments

Example: HintView, the HINT viewer.

• Backend:
  – literate cweb programs
• User Interface:
  – C (Windows)
  – Java (Android)
• Rendering:
  – OpenGL
Why?

Why do we want cweb Source Code of \( \text{T}_{\text{E}}\text{X} \) ?

- Toolchain
- Modification
- Debugging
- Education
- Experiments

HM
Why?

Why do we want cweb Source Code of \( \text{T}_\text{E} \text{X} \) ?

• Toolchain
• Modification
• Debugging
• Education
• Experiments
How?

• Start with tex.web

• Apply web change files, e.g. $\varepsilon^{-T_E}X$

• Apply patch files

• Convert web file to cweb file
  – run web2w
  – apply patch file

• Apply cweb change files, e.g. for $kT_E X$

• For $T_E X$ related projects
  – Extract code and Link
How?

- Start with `tex.web`
- Apply web change files, e.g. $\varepsilon$-TEX
- Apply patch files
- Convert web file to cweb file
  - run `web2w`
  - apply patch file
- Apply cweb change files, e.g. for kTEX
- For TEX related projects
  - Extract code and Link

The basis: `ctex.w`

- `tex.web`
- `web2w` systematic changes
- `ctex.patch`
- `ctex.w` individual changes

- Close to Don Knuth's original

TEX
How?

- Start with tex.web
- Apply web change files, e.g. $\varepsilon$-\TeX
- Apply patch files
- Convert web file to cweb file
  - run web2w
  - apply patch files
- Apply cweb change files, e.g. for k\TeX
- For \TeX related projects
  - Extract code and Link

The extended basis: ectex.w

- $\varepsilon$-\TeX
- \TeX
- \TeX

systematic changes

individual changes

- Extended features of $\varepsilon$-\TeX
How?

• Start with tex.web
• Apply web change files, e.g. \( \varepsilon\text{-TEX} \)
• Apply patch files
• Convert web file to cweb file
  – run web2w
  – apply patch files
• Apply cweb change files, e.g. for \( k\text{-TEX} \)
• For \( \text{TEX} \) related projects
  – Extract code and Link

• Large constants, e.g. 64 bit memory words
• Extended features of \( \varepsilon\text{-TEX} \)
How?

- Start with `tex.web`
- Apply web change files, e.g. $\varepsilon$-TEX
- Apply patch files
- Convert web file to cweb file
  - run `web2w`
  - apply patch files
- Apply cweb change files, e.g. for kTEX
- For T\(\text{E}\)X related projects
  - Extract code and Link

\(\text{T\(\text{E}\)X Live compatible T\(\text{E}\)X engine}\)
**What?**

**systematic changes: Macros**

- web2w translation of WEB macros to cweb macros

WEB:

@d define(#) == if global then geq_define(#) @+ else eq_define(#)

...

@<Assignments@> =

set_font: define(cur_font_loc, data, cur_chr);


cweb:

@d define(A, B, C) if (global) geq_define(A, B, C); @+ else eq_define(A, B, C)
What?

**systematic changes: Macros**

- New in web2w version 1.0

**WEB:**

```latex
@d \text{char\_info\_end}(#)\quad=\quad##]qqqq
@d \text{char\_info}(#)\quad=\quad\text{font\_info[char\_base[#]+char\_info\_end}

... \quad\text{cur\_h:=cur\_h+char\_width}(f)(\text{char\_info}(f)(c));
```

**cweb:**

```latex
@d \text{char\_info}(A, B)\quad\text{font\_info[char\_base[A]+B].qqqq}

... \quad\text{cur\_h=cur\_h+char\_width}(f, \text{char\_info}(f, c));
```
**What?**

*individual changes: nonlocal goto*

```c
@@<Error hand...@>
static void jump_out(void)
- {@+goto end_of_TEX;
+ {@+ close_files_and_terminate(); exit(0);
 }
```
What?

*individual changes: cwebmac.tex has no \L*

\<Types...\>=

typedef int scaled; /*this type is used for scaled integers*/
- typedef int32_t nonnegative_integer; /*$0 \L x < 2^{31}$*/
+ typedef int32_t nonnegative_integer; /*$0 \le x < 2^{31}$*/
  typedef int8_t small_number; /*this type is self-explanatory*/
What?

*individual changes: single character strings*

@<Put each...@>=

- primitive('-', discretionary, 1);
+ primitive("-", discretionary, 1);

@!@:Single-character primitives -}{{

\quad\{\\-\}@>
What?

*individual changes: use / as directory separator*

- @d TEX_area "TeXinputs:"
+ @d TEX_area "TeXinputs/"
  @.TeXinputs@

- @d TEX_font_area "TeXfonts:"
+ @d TEX_font_area "TeXfonts/"
  @.TeXfonts@