New ČSplain of 2012

Petr Olšák

The ČSplain package has existed since 1994 and it is a gentle extension of plain \TeX{} to make using Czech and Slovak languages feasible. This was the case until October 2012, when the author carried out significant revisions and additions to ČSplain. The basic change resulted from the decision to set the default input encoding of ČSplain to UTF-8. In addition, ČSplain got many other new features: the possibility of loading all available hyphenation patterns, the ability to cooperate with 16-bit \TeX{} engines (Lua\TeX{}, Xe\TeX{}), more effective work with fonts including math, easy switching of the internal encoding (including Unicode), and the user-friendly macros OPmac.

In the default configuration, ČSplain remains a gentle extension of plain \TeX{}, backwards-compatible with previous versions. The new possibilities are easily accessed with \texttt{\input} and when they are used it is no longer correct to talk of a gentle extension. On the contrary, it is a strong competitor to all other macro systems based on \TeX{}, even very large ones. ČSplain has advantages in its simplicity, effectiveness, and ease of usage.

The new ČSplain is available through CTAN and the usual \TeX{} distributions, and its home on the web is \url{http://petr.olsak.net/csplain-e.html}.

Introduction

In October 2012, a discussion was held on the cstex@ mailing list about the configuration of the input encoding of ČSplain. It was shown that for many years ČSplain used the wrong default input encoding on MS Windows: ISO 8859-2, which is foreign on this operating system. I was surprised.

Our old decision was that the input encoding of ČSplain was to be set depending on the operating system in use. This is similar to the ASCII versus EBCDIC encodings on old systems, where \TeX{} did reencoding of its input depending on its environment. It is essential that when the Czech and Slovak characters in the source file are shown correctly in the text editor then ČSplain prints them correctly too. On the other hand, when we see bad characters in the text editor, we cannot wonder that ČSplain produces broken output. Unfortunately, this idea was valid ten years ago, but not so much today. Nowadays there are text editors with special intelligence—they try to autodetect the encoding and they try to show anything properly. In such an environment, the above rule makes no sense. These modern editors handle the UTF-8 encoding, so we decided that this will be implicitly set as the input encoding of ČSplain on all systems.

The conversion between UTF-8 input codes and the internal encoding (i.e. font encoding and hyphenation pattern encoding) must be done straightforwardly at the input processor level. No active characters are allowed for this purpose. When we do
\begin{verbatim}
def\test\#1\#2\%
  {the first character is \#1, second is \#2}
test \&
\end{verbatim}
then we expect the output “the first character is \v{c}, second is \v{r}”. Therefore, ČSplain needs to activate the enc\TeX{} extension in 8-bit \TeX{} engines (\TeX{}, pdf\TeX{}). The 16-bit \TeX{} engines are more straightforwardly used for this case.

Format generation

The following lines show various methods to generate the format files csplain and pdftcsplain. The implicit output (DVI and PDF) is set by the name of generated format (csplain sets DVI output, while pdftcsplain sets PDF output).

\begin{verbatim}
pdftex -ini -enc "\let\enc=\input csplain.ini"
pdftex -jobname csplain -ini -etex \ -enc csplain-utf8.ini pdftex -jobname pdftcsplain -ini -etex \ -enc csplain-utf8.ini xetex -jobname pdftcsplain -etex -ini csplain.ini lualatex -jobname pdftcsplain -ini csplain.ini
\end{verbatim}

ČSplain — basic features

The basic behavior of ČSplain is similar to plain \TeX{}. The only difference is that the default \texttt{\hspace{}} and \texttt{\vspace{}} are set to create one inch margins in A4 paper format, not letter format. One can consider that the second difference is the presence of macros unknown in plain \TeX{}:

\begin{verbatim}
\chyp\ % Czech hyphenation patterns and
  \Frenchspacing initialised.
\shyp\ % Slovak hyphenation patterns and
  \Frenchspacing initialised.
\csaccents\ % redefines \textbackslash \v{c} \textbackslash \v{r} \textbackslash \v{r} \textbackslash \v{r} \textbackslash \v{r} \textbackslash \v{r} \textbackslash \v{r} \textbackslash \v{r} \textbackslash \v{r}
  \% to expand to given internal slot.
\end{verbatim}

You can return to the default behavior with:

\begin{verbatim}
\ehyp\ % US hyphenation patterns and
  \nonfrenchspacing.
\cmaccents\ % \textbackslash \v{c}, \textbackslash \v{r} etc. expand to
  \\accent\ primitive.
\end{verbatim}

The implicit internal encoding and the implicit fonts are set to \texttt{\encoding/\fonts} in ČSplain. It
means that (for example) the font \texttt{csr10} is preloaded as \texttt{tenrm} instead of \texttt{cmr10}. These \texttt{cs*} fonts keep the 7-bit half of the encoding table the same as their \texttt{cm*} counterparts, while Czech and Slovak letters are placed in the second part of encoding table, ordered by ISO-8859-2.

\texttt{C\textsc{s}plain} defines control sequences which correspond to the special glyphs used in \texttt{C\textsc{s}fonts}.

\begin{verbatim}
% left Czech double quote.
\clqq 
% right Czech double quote.
\crqq 
% left French double quote.
\flqq % (used at right side in Czech).
% right French double quote.
\frqq % (used at left side in Czech).
% per mille character.
\promile
% quotation macro: \uv{text} gives \clqq text\crqq.
% Polish a-ogonek (composed from components).
\ogonek a
\end{verbatim}

\textbf{UTF-8 input encoding when \texttt{encT\textsc{e}X} is used}

You can recognize the UTF-8 encoded \texttt{C\textsc{s}plain} with \texttt{encT\textsc{e}X} by the message:

\texttt{The format: csplain <Nov. 2012>}.  
\texttt{The cs-fonts are preloaded and A4 size implicitly defined.}
\texttt{The utf8->iso8859-2 re-encoding of Czech+Slovak alphabet activated by encTEX}

Many thousands of character codes can occur in UTF-8 input, but by default, \texttt{C\textsc{s}plain} is able to read only characters from ASCII and the Czech and Slovak alphabets:

\begin{verbatim}
Á á Æ æ Í í Ý ý Ž ž
\end{verbatim}

These characters are mapped by \texttt{encT\textsc{e}X} to one byte (one slot) corresponding to the internal encoding. Moreover, the characters known from plain \TeX\ are mapped to the control sequences:

\begin{verbatim}
\ss \ß, \l, \L, \ae \æ, \oe \ø, \AE \Æ, \OE \Œ, \o ø, \O Ø, \i i \i j, \aa \å, \AA \Å, \S §, \P ¶, \copyright ©, \dots ..., \dag ¶, \ddag ‡.
\end{verbatim}

Enc\TeX\ is able to map the UTF-8 code to the internal 8-bit slot or to the control sequence. When such a mapped control sequence or internal 8-bit slot is processed by the \texttt{\write} primitive, it is converted back to the UTF-8 code. So, the 8-bit \TeX\ engine can handle an unlimited number of UTF-8 codes. But by default, only the characters mentioned above are properly processed by \texttt{C\textsc{s}plain}. If another UTF-8 code occurs in the input, \texttt{C\textsc{s}plain} reports the following warning (the \texttt{N} character is used in this example):

\begin{verbatim}
WARNING: unknown UTF-8 code: ‘˜N = ^^c3^^91’ (line: 42)
\end{verbatim}

and users can add their own mapping and definition of such a character. For example:

\begin{verbatim}
\mubyte\Ntilde ^^c3^^91\endmubyte % \UTF-8 code mapped to \Ntilde.
\def\Ntilde{\~N} % The \Ntilde is defined.
\end{verbatim}

Now \texttt{C\textsc{s}plain} processes the \texttt{N} character properly even though it is not included in the Czech or Slovak alphabets.

The distribution \texttt{enctex.tar.gz} contains these two files:

\begin{verbatim}
utf8lat1.tex % Latin1 Supplement U+0080-U+00FF
utf8lata.tex % Latin Extended-A U+0100-U+017F
\end{verbatim}

These files do the mapping of the abovementioned UTF-8 codes by \texttt{encT\textsc{e}X} and provide the definitions for the mapped control sequences. You can \texttt{\input} them to your document and/or create analogous files for your purposes.

\section*{Internal encoding}

The internal encoding means the encoding of the fonts and hyphenation patterns that are used. By default, \texttt{C\textsc{s}plain} sets the internal encoding to the \texttt{CS}-encoding (as mentioned above). But you can change this encoding via \texttt{\input} at the beginning of your document. There are two possibilities:

\begin{verbatim}
\input t1code % the T1 internal encoding is set
\input ucode % the Unicode internal encoding % is set (in 16-bit \TeX\ engines)
\end{verbatim}

These \texttt{\input} files do the following:

\begin{itemize}
\item Set the correct \texttt{\uccode/\lccode}.
\item Reset the \texttt{\chph} and \texttt{\shph} macros, so they choose the hyphenation patterns in proper encoding.
\item Remap the UTF-8 codes to the new slots, if \texttt{encT\textsc{e}X} is used.
\item Redefine some character-like control sequences (\texttt{\ss}, etc.).
\item Redefine \texttt{\csaccents}, so \texttt{\v x}, \texttt{\v x}, etc. expand to the right slots.
\end{itemize}

As you can see, these files don’t reload the fonts with the proper encoding. This has to be done with the next \texttt{\input} in your document, for example \texttt{\input lmfonts} or \texttt{ctimes} or \texttt{cs-pagella}.

\texttt{C\textsc{s}plain} preloads the Czech and Slovak hyphenation patterns in \texttt{CS}-encoding, in T1 encoding and

Petr Olšák
(if a 16-bit \TeX{} engine is detected) in Unicode. The only thing the user need be concerned with is initializing the hyphenation patterns with \chyrph or \shyph after the \input tcode or \input ucode is done. The section below, “More languages”, describes how \Cs{plain} is able to load hyphenation patterns of another languages.

**Font handling**

\Cs{plain} introduces a simple font-resizing principle. The main credo is: “power is in simplicity”. That is the reason why I don’t use NFFS, for example.

The command \font\foo=something declares font selector \foo which selects the font something. The terminology font selector in this section is used only for selectors declared by the \font primitive. This means that \bf (for example) isn’t a font selector. It is a macro.

\Cs{plain} defines the following macros for font size handling.

- \resizefont\foo resizes the font represented by font selector \foo. More precisely, it declares (locally) \foo as the same font but with the size given in the macro \sizespec. The \sizespec macro can have the form at(dimen) or scale(factor).
- \regfont\foo registers the font selector \foo as a resizable font. By default \Cs{plain} declares the following selectors with \regfont: \tenrm, \tenit, \tenbf, \tenbi and \tenbt. Users can declare more selectors.
- \resizeall resizes (locally) all registered font selectors to the size given by the \sizespec macro.
- \letfont \foo=\bar at(dimen) or \letfont \foo=\bar scaled(factor) declares a new font selector \foo as the same font as \bar with the given size. The \bar font selector is unchanged.

Here’s an example:

\input chelvet \tenrm
\font\zapfchan=pzcmi8z \regfont\zapfchan
\def\sizespec{at13.5pt} \resizeall \tenrm
\baselineskip=15pt

Here is the typesetting at size 13.5pt including \{\it italics\}, \{\bf bold\} and including the \{\zapfchan Zapf Chancery font\}.

\input ctimes \tenrm
\def\sizespec{at8pt} \resizeall \tenrm

Now all the typesetting is at the 8pt size.

Another example uses the font loading files:

\input chelvet \tenrm, \tenit, etc. is now \tenrm.
\letfont\titlefont = \tenbf at14.4pt
\input ctimes \tenrm, etc. is Times Roman.
\def\sizespec{at11pt}\resizeall \tenrm
\baselineskip=15pt
\input chelvet \tenrm\tenit, \tenbf, \tenbi and \tenbt.
\input ctimes \tenrm\tenit, \tenbf, \tenbi and \tenbt.
\input ctimes \tenrm\tenit, \tenbf, \tenbi and \tenbt.

The \tenrm macro switches the whole family of TimesRoman to the 9pt size, e.g., for footnotes.

Note #1. The font selectors \tenrm, \tenit, etc. have the subword \ten in its name but this is only for historical reasons. The current meaning of these selectors can be fonts at an arbitrary size.

Note #2. These macros do not solve the resizing of math fonts. This is the subject of the following section.

Note #3. The selection of the proper design size (\cmr5 or \cmr7 or … or \cmr17) is not solved by default. But the math font macros solve this and you can simply redefine \resizefont so that the proper design size is selected.
Math fonts

The \texttt{CSplain} package provides two macro files for math fonts: \texttt{ams-math.tex} and \texttt{tx-math.tex}. The first one loads \LaTeX{} fonts and declares hundreds of math symbols and operators like \LaTeX{}. The second macro file does the same but loads the \TeX{} fonts which are visually compatible with Times Roman and similar designs.

By default, neither of these macro files are read. But you can load \texttt{ams-math.tex} explicitly, or the proper macro file is loaded implicitly with \texttt{\input ctimes, lfonts, etc.}

These files provide the macro:
\begin{verbatim}
\setmathsizes[(text)/(script)/(scriptscript)]
\end{verbatim}

in which the user can set the sizes of basic text, script and superscript. The parameters have to be written without unit (the unit pt is used). For example \texttt{\setmathsizes[10/7/5]} is the default from \texttt{plain TeX}.

The following math alphabets are available after \texttt{ams-math.tex} or \texttt{tx-math.tex} is loaded:
\begin{verbatim}
\mit % mathematical variables
\rm, \it % text fonts in math
\bf, \bi % bold sans fonts (might be different than text fonts)
\cal % normal calligraphic
\script % script
\frak % fraktur
\bbchar % double stroked letters
\end{verbatim}

The \texttt{ams-math.tex} defines the \texttt{\reftfm} macro to declare the mapping from a desired size to the list of design sizes represented by names of the metric files. For more information about this, see the file \texttt{ams-math.tex}, where \texttt{\reftfm} is defined and used. Once this mapping is set, you can redefine the internal subpart of the \texttt{\resizesize} macro in the following way:
\begin{verbatim}
\def\resizesizetimeout &1 \#2relax
{\whichtfm[\#1] \sizespec\relax}
\end{verbatim}

Now \texttt{\resizesize} chooses the right metrics if \texttt{\sizespec} and \texttt{\dgsizes} are properly set. This complexity can be hidden from the user, if he or she uses the \texttt{\typosize} and \texttt{\typoscale} macros from OPmac.

The following example shows how to set the font for a title that includes math formulas:
\begin{verbatim}
\def\titlefont{\def\at14pt\resizesize\tenbf
\tenbf \setmathsizes[14/9.8/7] \boldmath}
\def\titlefont{\uppar{\centerline{\titlefont \#1}}}
\end{verbatim}

\texttt{\titlefont More about $\int_{-\infty}^{\infty} f(t)\{\text{rm d}\} t$}

The \texttt{\boldmath} command selects the alternative set of all math families more compatible with \textbf{bold} fonts usually used in titles.

Unicode fonts

Historically, \texttt{CSplain} worked with 8-bit \TeX{} engines where Unicode fonts are impossible. So, all the font handling mentioned so far is primarily intended for 8-bit fonts. The Unicode support for text fonts in \texttt{CSplain} is only experimental, and Unicode math isn’t solved in \texttt{CSplain} at all.

The 16-bit \TeX{} engines expect the UTF-8 input encoding and work in Unicode internally. So T1-encoded fonts cannot be used because Czech and Slovak alphabets are unfortunately not in the intersection of T1 and Unicode encodings. On the other hand, colleagues writing in German or French can use T1-encoded 8-bit fonts in 16-bit \TeX{} engines because their whole alphabet is in this intersection.

\texttt{XeTeX} has a font loader linked with system libraries and it extends the syntax of the \texttt{\font} primitive. For example:
\begin{verbatim}
\font\foo=[(filename)]:\langle\fontfeatures\rangle\ (sizespec)
\end{verbatim}

where \texttt{(filename)} is the file name without the .otf suffix and the \texttt{(sizespec) is at(dimen) or scaled \langle\factor\rangle}. The \texttt{\fontfeatures} are font modifiers separated by semicolon. You have to know which features are implemented in the font and which in the font loader. For example, \texttt{XeTeX}’s font loader provides the feature \texttt{mapping=tex-text} which activates the usual \TeX{} ligatures like \texttt{\rightarrow}. The normal ligatures (e.g., ‘fi’) are activated implicitly.

On the other hand, Lu\TeX{} implements its extension of the font loader by Lua code. I have extracted the core of this code (from \texttt{luatotload sty}) for \texttt{CSplain}, in a file \texttt{luafonts.tex}. Its stability can’t be guaranteed because the Lua functions from the Lu\TeX{} distribution are called, and they may change in the future. If Lu\TeX{} is being used, the files \texttt{lfonts.tex, cs-termes.tex, cs-heros.tex}, etc. input \texttt{luafonts.tex} before the first usage of the extended \texttt{\font} primitive.

The extension of the \texttt{\font} primitive seems to have the same syntax in \texttt{XeTeX} and Lu\TeX{}. But, unfortunately, the font features are different. By default, no ligatures are activated in Unicode fonts in Lu\TeX{}. Users must use \texttt{\script=latn} to activate the fi-ligatures and \texttt{\ttlig} to activate the \texttt{TEx} special ligatures. Users can define the \texttt{\fontfeatures} macro for special needs of features. If this macro isn’t defined, \texttt{CSplain}’s font-loading macros make the following default:

Petr Olšák
\def\fontfeatures
\{mapping=tex-text;script=latn;+tlig\}
which works in both \TeX\ and \LaTeX.\n
\section*{More languages}

The following hyphenation patterns are preloaded in \texttt{CSplain} by default:

- \texttt{\USenglish=0} \ldots default US hyphenation patterns from plain \TeX, ASCII encoding.
- \texttt{\czILtwo=5} \ldots Czech patterns, ISO-8859-2.
- \texttt{\skILtwo=6} \ldots Slovak patterns, ISO-8859-2.
- \texttt{\czCork=15} \ldots Czech patterns, T1 encoding.
- \texttt{\skCork=16} \ldots Slovak patterns, T1 encoding.
- \texttt{\czUnicode=115} \ldots Czech patterns, Unicode (only for 16-bit \TeX\ engines).
- \texttt{\skUnicode=116} \ldots Slovak patterns, Unicode (only for 16-bit \TeX\ engine).

Hyphenation patterns are selected with \texttt{\uslang}, \texttt{\czlang} and \texttt{\sklang}, which are equivalent to the old selectors \texttt{\ehyph}, \texttt{\chyph} and \texttt{\shyph}. The proper encoding is used if the command \texttt{\input t1code} or \texttt{\input ucode} precedes the patterns selector.

Since 2012, \texttt{CSplain} is able to load hyphenation patterns of other languages (ca. 50 languages). If the patterns use a subset of T1 encoding, they can be loaded in T1 (alias Cork) and/or in Unicode. Otherwise, only the Unicode encoding for the patterns is allowed. Unicode patterns can be loaded only in 16-bit \TeX\ engines.

The loading of extra hyphenation patterns can be done on the command line when format is generated. Examples follow:

\begin{verbatim}
  pdftex -ini -enc \ 
    "\let\plCork=y \let\enc=u \input csplain.ini"
  pdftex -ini -enc "\let\allpatterns=y \let\enc=u \input csplain.ini"
  luatex -jobname pdfcsplain -ini \ 
    "\let\ruUnicode=y \input csplain.ini"
  luatex -jobname pdfcsplain -ini "\let\allpatterns=y \input csplain.ini"
\end{verbatim}

The first line adds Polish hyphenation patterns in the T1 encoding to \texttt{CSplain}. The second line loads all available hyphenation patterns for 8-bit \TeX\ engines (i.e. Czech&Slovak in ISO-8859-2 and T1, and others, ca. 30 languages, in T1). The third line loads the Russian hyphenation patterns in Unicode. Finally, the last line loads all available hyphenation patterns (in T1 and in Unicode). The pattern selectors have the form \texttt{\langle\texttt{twoletters}\rangle\texttt{lang}}, for example \texttt{\pllang}, \texttt{\delang}, \texttt{\itlang}, \texttt{\rulang} etc. Please read the \texttt{hyphen.lan} file for more information.

\section*{The OPmac macro package}

The OPmac (Olsak’s Plain macros) package is part of \texttt{CSplain}. It provides more \LaTeX-like features in plain \TeX: font size changing, automatic creation of tables of contents and indexes, working with bibliography databases, tables, references including hyperlinks options, etc. For more information about this macro package, see the companion article in this same issue of \textit{TUGboat}.

\begin{itemize}
  \item Petr Olšák \newline
  Czech Technical University \newline
  in Prague \newline
  Czech Republic
\end{itemize}

New \texttt{CSplain} of 2012