

milog.cls

v1.0

Eine \LaTeX -Klasse zur Erfüllung der Dokumentationspflichten nach dem Mindestlohngesetz (MiLoG)

2016/07/01

Package author:
Josef Kleber

1	Klassenoptionen	4
2	Die Konfigurationsdatei milog.cfg	4
3	Das CSV-Datenformat	4
4	Befehle	5
4.1	\milog	5
4.2	\Formular	5
4.3	\milogsetup	5
5	Beispiel	6
6	An app a day keeps the work away	6
7	Implementierung	7
7.1	milog.cls	7
7.2	milog	13
8	Literatur	17
9	Change History	18
10	Index	19

Zusammenfassung

Seit dem 1. Januar 2015 gilt in Deutschland grundsätzlich für alle Arbeitnehmer ein flächendeckender gesetzlicher Mindestlohn in Höhe von derzeit 8,50€ pro Stunde. Mit Wirkung ab 1. August 2015 wurden die Dokumentations- und Aufzeichnungspflichten gelockert. Nach §17 MiLoG, muss Beginn, Ende und Dauer der täglichen Arbeitszeit der in §22 MiLoG definierten Arbeitnehmern (formlos) aufgezeichnet werden. Zusätzlich ermöglicht milog.cls aus praktischen Gründen die Aufzeichnung von unbezahlten Pausen und Bemerkungen (Ruhetag, Urlaub, Krank, ...).

Die Erfassung der Arbeitszeiten erfolgt in einer simplen CSV-Datei, aus der die Klasse automatisch einen Arbeitszeitnachweis erstellt. Alternativ können die Daten auch durch einen CSV-Export — mit eventueller Nachbearbeitung — einer geeigneten App erhoben werden.

The milog.cls class provides means to fulfill the documentation duties by the German minimum wage law MiLoG. The recording of working hours is carried out in a simple CSV file from which the class will automatically create a time sheet. Alternatively, data can also be collected by a CSV export of a suitable app.

1 Klassenoptionen

Grundsätzlich handelt es sich bei `milog.cls` um eine Wrapper-Klasse um `article.cls`, d. h. es können grundsätzlich die „normalen“ Klassenoptionen verwendet werden. Dies sollte jedoch nicht nötig sein, da diese Klasse ein bestimmtes Formular-Design umsetzt! Sie bietet daher die folgenden Optionen, die auch mit lokaler Wirkung im `\milog` Makro angewendet werden können:

`heading` [`Arbeitszeitnachweis nach dem Mindestlohngesetz (§17 MiLoG)`] Überschrift

`cc` [`false`] erstellt – falls auf `true` gesetzt – einen Durchschlag mit dem Wasserzeichen „Kopie“

`companyname` [`Firma`] Bezeichnung der Arbeitgeberorganisation

`company` [] Firmenname

`namename` [`Name, Vorname`] Bezeichnung der Personenidentifikation

`name` [] Name

`idname` [`Geburtsdatum, Personalnummer`] Bezeichnung einer zusätzlichen Personenidentifikation

`id` [] Zusatz-Id

`monthname` [`Monat/Jahr`] Bezeichnung der Aufzeichnungsperiode

`month` [] Aufzeichnungsperiode

`color` [`gray!40`] Farbvariation der Aufzeichnungsliste

2 Die Konfigurationsdatei `milog.cfg`

Falls vorhanden, wird die Konfigurationsdatei `milog.cfg` eingebunden. Darin können etwa das Festlegen neuer Defaults für die oben angegebenen Optionen (siehe Abschnitt 4.3 auf Seite 5) oder auch Redefinitionen des Layouts ausgelagert werden. Dies kann sowohl systemweit in einem (lokalen) $\text{T}_{\text{E}}\text{X}$ -Tree erfolgen oder auch lokal im aktuellen Arbeitsverzeichnis!

3 Das CSV-Datenformat

Das folgende Beispiel zeigt den Aufbau der CSV-Dateien für den Arbeitszeitnachweis. Die einzelnen Spalten sind dabei durch Spaces oder Tabulatoren zu trennen. Dadurch müssen „leere“ Spalten durch die speziellen Marker `{}` bzw.

nan gekennzeichnet werden. Diese werden in der ausgegebenen Tabelle durch „-“ bzw. „\space“ ersetzt.

tag	beginn	pause	ende	dauer	bemerkung
01	06:20	{}	12:00	05:40	{}
02	{}	{}	{}	{}	Ruhetag
03	{}	{}	{}	{}	Ruhetag
04	06:20	{}	12:00	05:40	{}
05	06:20	{}	12:00	05:40	{}
06	06:20	{}	12:05	05:45	{}
07	06:20	{}	12:10	05:50	{}
08	06:20	{}	12:05	05:45	{}
09	{}	{}	{}	{}	Ruhetag
10	{}	{}	{}	{}	Ruhetag
nan	nan	nan	nan	32:20	nan

Eine Besonderheit stellt die letzte Zeile dar, in der die Gesamtarbeitszeit angegeben werden kann. Die Abschlusslinie der Tabelle wird **vor** der letzten Zeile gesetzt. Es muss daher eine letzte Zeile mit nan-Einträgen vorhanden sein, falls die Ausgabe der Gesamtarbeitszeit nicht gewünscht sein sollte.

4 Befehle

4.1 \milog

`\milog` [*Optionen*]{*CSV-Datei*} Das `\milog` Makro liest die {*CSV-Datei*} ein und erstellt daraus automatisch den gewünschten Arbeitszeitnachweis.

4.2 \Formular

`\Formular` Das `\Formular` Makro stellt ein Blankoformular in Form einer „inline table“ zur Verfügung. Es kann anstatt einer CSV-Datei im `\milog` Makro verwendet werden.

4.3 \milogsetup

`\milogsetup`{*Optionen*} Das `\milogsetup` Makro dient dem erneuten Setzen der globalen Optionen. Insbesondere in Verbindung mit der Konfigurationsdatei `mi log . c f g` (siehe Abschnitt 2 auf Seite 4) zum Setzen systemweiter oder auf das aktuelle Verzeichnis bezogene Defaults.

5 Beispiel

Im folgenden Beispiel setzen wir die persönlichen Angaben in der Konfigurationsdatei `milog.cfg`.

```
1 \milogsetup{company={\TeX\ Satz GmbH},name={Mustermann, Max},
2           id={01.01.1970/0815}}
```

Anstatt in der Konfigurationsdatei könnten die persönlichen Angaben auch als Klassenoptionen gesetzt werden. Den Zeitraum des Arbeitszeitnachweises setzen wir als lokale Option im `\milog` Makro. Außerdem erzwingen wir durch Setzen der Option `cc` einen Durchschlag.

```
1 \documentclass{milog}
2 \usepackage[utf8]{inputenc}
3 \usepackage[T1]{fontenc}
4 \usepackage{libertine}
5 \begin{document}
6 \milog[month={05/2016},cc=true]{201605.csv}
7 \end{document}
```

6 An app a day keeps the work away

Abschließend möchte ich noch eine Möglichkeit zur weitestgehend automatischen Erstellung des Arbeitszeitnachweises aus dem CSV-Export einer Arbeitszeit-App aufzeigen. Exemplarisch nutzen wir hierzu die Android-App „Arbeitszeiterfassung“ [2]. Dazu wandeln wir den CSV-Export der App mit Hilfe des `milog` Bash-Skripts (siehe Abschnitt 7.2 auf Seite 13) in das von `milog.cls` benötigte CSV-Format (siehe Abschnitt 3 auf Seite 4). Grundsätzlich ermöglicht die App das Erfassen von zwei Schichten. Zur Vereinfachung wertet das Skript jedoch nur die erste Schicht aus.

```
1 $ milog -i 201605.dat
2 reading 201605.dat
3 writing _milog_.csv
4 creating milog.pdf
```

Insbesondere verwenden wir für den Namen der CSV-Exportdatei das Format `YYYYMM`. Das Skript setzt damit die `month` Option automatisch. Die Angaben zur Person setzen wir mit Hilfe des `\milogsetup` Makros (siehe Abschnitt 4.3 auf Seite 5) in der Konfigurationsdatei `milog.cfg` (siehe Abschnitt 2 auf Seite 4). Alternativ wäre auch eine Anpassung der HERE-Datei innerhalb des Skripts möglich. Die Verwendung der Konfigurationsdatei im jeweiligen Arbeitsverzeichnis ermöglicht jedoch die Erstellung eines Arbeitszeitnachweises

für unterschiedliche Personen mit demselben Script.

Falls Anpassungen der CSV-Datei — etwa in der Spalte bemerkung — nötig sein sollten, kann die Erstellung der PDF-Datei in zwei Schritten erfolgen!

```

1 $ milog -i 201605.dat -o 201605.csv -c
2 reading 201605.dat
3 writing 201605.csv
4
5 [ ... Bearbeitung der CSV-Datei ... ]
6
7 $ milog -i 201605.csv -p 201605 -l
8 creating 201605.pdf

```

7 Implementierung

7.1 milog.cls

```
1 <class>
```

Zunächst stellen wir die L^AT_EX Klasse `milog.cls` bereit.

```
2 \NeedsTeXFormat{LaTeX2e}%
3 \ProvidesClass{milog}[2016/07/01 class for MiLoG documentation v1.0]%
```

Wir laden die Pakete `xkeyval` und `xkvlxp` um geschweifte Klammern und etwas mehr in den globalen Klassenoptionen zu erlauben.

```
4 \RequirePackage{xkeyval}%
5 \RequirePackage{xkvlxp}%
```

Wir stellen das Makro `\MLG@JK@define@key` zur Verfügung, das die Klassenoptionen mit globaler Wirkung und Optionen für das `\milog` Makro mit lokaler Wirkung definiert. Es bietet vier Argumente $\{\langle prefix \rangle\}$, $\{\langle class \rangle\}$, $\{\langle option \rangle\}$ and $\{\langle default \rangle\}$.

```
6 \newcommand*\MLG@JK@define@key[4]%
7 {%
8   \expandafter\gdef\csname#1@#3\endcsname{#4}%
9   \define@key{#2.cls}{#3}[#4]%
10  {%
11   \expandafter\gdef\csname#1@#3\endcsname{##1}%
12  }%
13  \define@key{#2}{#3}%
14  {%
15   \expandafter\def\csname#1@#3\endcsname{##1}%
16  }%
17 }%
```

Damit können wir nun unsere Optionen definieren.

```

18 \MLG@JK@define@key{MLG@JK}{milog}{color}{gray!40}%
19 \MLG@JK@define@key{MLG@JK}{milog}{companyname}{Firma}%
20 \MLG@JK@define@key{MLG@JK}{milog}{company}{}%
21 \MLG@JK@define@key{MLG@JK}{milog}{namenname}{Name, Vorname}%
22 \MLG@JK@define@key{MLG@JK}{milog}{name}{}%
23 \MLG@JK@define@key{MLG@JK}{milog}{idname}{Geburtsdatum/Personalnummer}%
24 \MLG@JK@define@key{MLG@JK}{milog}{id}{}%
25 \MLG@JK@define@key{MLG@JK}{milog}{monthname}{Monat/Jahr}%
26 \MLG@JK@define@key{MLG@JK}{milog}{month}{}%
27 \MLG@JK@define@key{MLG@JK}{milog}{heading}{Arbeitszeitznachweis nach %
28 dem Mindestlohngesetz (\S 17 MiLoG)}%
29 \MLG@JK@define@key{MLG@JK}{milog}{cc}{false}%

```

Jetzt wenden wir die Klassenoptionen an und setzen damit die Optionsmakros mit den Default-Werten.

```

30 \DeclareOptionX*{\PassOptionsToClass{\CurrentOption}{article}}%
31 \ExecuteOptionsX{color, companyname, company, namenname, name, idname, id, %
32                 monthname, month, heading, cc}%
33 \ProcessOptionsX*\relax%

```

Wir laden die `article.cls` Klasse und müssen daher nichts weiter definieren.

```

34 \LoadClass{article}%

```

Wir laden weitere benötigte Pakete.

```

35 \RequirePackage[margin={2cm, 2cm}]{geometry}%
36 \RequirePackage{tabularx}%
37 \RequirePackage{booktabs}%
38 \RequirePackage{colortbl}%
39 \RequirePackage{xifthen}%
40 \RequirePackage[pages=some]{background}%
41 \RequirePackage{transparent}%
42 \RequirePackage{pgfplotstable}%

```

Für unser „Formular“ benötigen wir keinen Absatzeinzug und keine Seitenzahlen.

```

43 \setlength{\parindent}{0pt}%
44 \pagestyle{empty}%

```

Definitionen: Überschrift

```

45 \newcommand*\MLG@JK@heading@font@size{\Large}%
46 \newcommand*\MLG@JK@heading@font@style{\bfseries}%

```

Definitionen: Abstände der Formularelemente

Durch die Verwendung von `\vfill` wird der verbleibende Abstand gleichmäßig verteilt, insbesondere bei unterschiedlich langen Monaten. Bei kürzeren Zeitabschnitten (wöchentlich) evtl. auf feste Abstände umdefinieren.

```

47 \newcommand*\MLG@JK@idtab@topskip{\vfill}%
48 \newcommand*\MLG@JK@csvtab@topskip{\vfill}%
49 \newcommand*\MLG@JK@signaturebox@topskip{\vfill\vfill}%

```


Definitionen: Tabelle (persönliche Daten)

```
50 \newcommand*\MLG@JK@idtab@width{13cm}%
51 \newcommand*\MLG@JK@idtab@as{1.3}%
```

Definitionen: Tabelle (Arbeitszeitnachweis aus csv)

```
52 \newcommand*\MLG@JK@csvtab@as{1.15}%
53 \newcommand*\MLG@JK@csvtab@opacity{0.5}%
```

Definitionen: cc (Optionen für \backgroundsetup)

```
54 \newcommand*\MLG@JK@bgopt@contents{Kopie}%
55 \newcommand*\MLG@JK@bgopt@color{gray!30}%
56 \newcommand*\MLG@JK@bgopt@opacity{0.5}%
57 \newcommand*\MLG@JK@bgopt@scale{25}%
```

Definitionen: Unterschriftenabschnitt

```
58 \newcommand*\MLG@JK@signaturebox@space{3cm}%
```

Definitionen: Unterschrift

```
59 \newcommand*\MLG@JK@signature@width{6cm}%
60 \newcommand*\MLG@JK@signature@labelskip{2ex}%
61 \newcommand*\MLG@JK@signature@employee{Arbeitnehmer}%
62 \newcommand*\MLG@JK@signature@employer{Arbeitgeber}%
```

Wir definieren die Ausgestaltung der Tabelle (csv). Insbesondere definieren wir die Ersetzungen „nan“ → „\space“ und „empty cells“ → „-“.

```
63 \pgfplotstableset%
64 {%
65   columns/tag/.style={string type,%
66                       column name={Tag},%
67                       string replace={nan}{\space}},%
68   columns/beginn/.style={string type,%
69                          column name={Arbeitsbeginn},%
70                          string replace={nan}{\space}},%
71   columns/pause/.style={string type,%
72                         column name={Unbezahlte Pause},%
73                         string replace={nan}{\space}},%
74   columns/ende/.style={string type,%
75                        column name={Arbeitsende},%
76                        string replace={nan}{\space}},%
77   columns/dauer/.style={string type,%
78                         column name={Arbeitsdauer},%
79                         string replace={nan}{\space}},%
80   columns/bemerkung/.style={string type,%
81                             column name={Bemerkung},%
82                             string replace={nan}{\space}},%
83   empty cells with={-},%
84   every head row/.style={before row=\toprule,after row=\midrule},%
85   every even row/.style={before row={\rowcolor{\MLG@JK@color}}},%
```

Wir helfen uns hier mit einem „dirty trick“! Die Abschlusslinie der Tabelle setzen wir **vor** der letzten Zeile! Damit kann und **muss** die letzte Zeile die Gesamtzahl der geleisteten Arbeitstunden bzw. „nan“ enthalten.

```
86 every last row/.style={before row=\bottomrule}%
87 }%
```

Wir definieren ein Makro, das ein colortbl-Makro ändert um Transparenz in farbigen Tabellenzeilen zu erlauben.

Siehe: <https://tex.stackexchange.com/questions/56424/alpha-channel-in-colortbl>

```
88 \newcommand*\MLG@JK@colortbl@transparent%
89 {%
90   \def\CT@do@color%
91   {%
92     \global\let\CT@do@color\relax%
93     \@tempdima\wd\z@%
94     \advance\@tempdima\@tempdimb%
95     \advance\@tempdima\@tempdimc%
96     \kern-\@tempdimb%
97     \transparent{\MLG@JK@csvtab@opacity}%
98     \leaders\vrule%
99     \hskip\@tempdima\@plus 1fill%
100    \kern-\@tempdimc%
101    \hskip-\wd\z@ \@plus -1fill%
102   }%
103 }%
```

`\Formular` Das Makro `\Formular` erlaubt das Setzen eines Blankoformulars für die Arbeitszeiterfassung.

`\Formular`

```
104 \pgfplotstableread{%
105 tag   beginn  pause   ende    dauer   bemerkung
106 1     nan      nan     nan     nan     nan
107 2     nan      nan     nan     nan     nan
108 3     nan      nan     nan     nan     nan
109 4     nan      nan     nan     nan     nan
110 5     nan      nan     nan     nan     nan
111 6     nan      nan     nan     nan     nan
112 7     nan      nan     nan     nan     nan
113 8     nan      nan     nan     nan     nan
114 9     nan      nan     nan     nan     nan
115 10    nan      nan     nan     nan     nan
116 11    nan      nan     nan     nan     nan
117 12    nan      nan     nan     nan     nan
118 13    nan      nan     nan     nan     nan
119 14    nan      nan     nan     nan     nan
120 15    nan      nan     nan     nan     nan
121 16    nan      nan     nan     nan     nan
122 17    nan      nan     nan     nan     nan
123 18    nan      nan     nan     nan     nan
```

```

124 19   nan   nan   nan   nan   nan
125 20   nan   nan   nan   nan   nan
126 21   nan   nan   nan   nan   nan
127 22   nan   nan   nan   nan   nan
128 23   nan   nan   nan   nan   nan
129 24   nan   nan   nan   nan   nan
130 25   nan   nan   nan   nan   nan
131 26   nan   nan   nan   nan   nan
132 27   nan   nan   nan   nan   nan
133 28   nan   nan   nan   nan   nan
134 29   nan   nan   nan   nan   nan
135 30   nan   nan   nan   nan   nan
136 31   nan   nan   nan   nan   nan
137 nan  nan   nan   nan   nan   nan
138 }\Formular%

```

Wir laden — falls vorhanden — die Datei `milog.cfg`

```

139 \AtBeginDocument%
140 {%
141   \IfFileExists{milog.cfg}%
142   {%
143     \input{milog.cfg}%
144   }%
145 }%

```

Wir setzen die persönlichen Daten innerhalb einer Tabelle mit erhöhtem Zeilenabstand.

```

146 \newcommand*\MLG@JK@idtab%
147 {%
148   \MLG@JK@idtab@topskip%
149   \renewcommand{\arraystretch}{\MLG@JK@idtab@as}%
150   \begin{tabularx}{\MLG@JK@idtab@width}{>\bfseries}r<{:}X}%
151   \MLG@JK@companyname & \MLG@JK@company \\%
152   \MLG@JK@namenname & \MLG@JK@name \\%
153   \ifthenelse{\equal{\MLG@JK@id}{}}%
154   {}%
155   {\MLG@JK@idname & \MLG@JK@id \\}%
156   \MLG@JK@monthname & \MLG@JK@month%
157   \end{tabularx}%
158 }%

```

Wir setzen die Arbeitszeiten (csv) innerhalb einer Tabelle mit leicht erhöhtem Zeilenabstand.

```

159 \newcommand*\MLG@JK@csvtab[1]%
160 {%
161   \MLG@JK@csvtab@topskip%
162   \renewcommand{\arraystretch}{\MLG@JK@csvtab@as}%
163   \pgfplotstabletypeset{#1}%
164 }%

```

Wir setzen Unterschriftslinien mit zentriertem Label.

```

165 \newcommand*\MLG@JK@signature[1]%
166 {%
167   \parbox{\MLG@JK@signature@width}%
168   {%
169     \dotfill\[\MLG@JK@signature@labelskip]%
170     \centering\footnotesize Datum, Unterschrift (#1)%
171   }%
172 }%
```

Wir setzen ein Unterschriftenblock für Arbeitnehmer und Arbeitgeber.

```

173 \newcommand*\MLG@JK@signaturebox%
174 {%
175   \MLG@JK@signaturebox@topskip%
176   \MLG@JK@signature{\MLG@JK@signature@employee}%
177   \hspace{\MLG@JK@signaturebox@space}%
178   \MLG@JK@signature{\MLG@JK@signature@employer}%
179 }%
```

Wir definieren ein Makro für den Aufbau des Arbeitszeitnachweises.

```

180 \newcommand*\MLG@JK@doc@body[1]%
181 {%
182   {\MLG@JK@heading@font@size\MLG@JK@heading@font@style\MLG@JK@heading}%
183   \MLG@JK@idtab%
184   \MLG@JK@csvtab{#1}%
185   \MLG@JK@signaturebox%
186 }%
```

`\milogsetup` Das Makro `\milogsetup` ermöglicht das erneute Setzen der globalen Optionen, z. B. in der Datei `milog.cfg`.

```
\milogsetup{<Optionen>}
```

```

187 \newcommand*\milogsetup[1]%
188 {%
189   \setkeys{milog.cls}{#1}%
190 }%
```

`\milog` Nun können wir das Usermakro `\milog` definieren

```
\milog[<Optionen>]{<CSV-Datei>}
```

```

191 \newcommand\milog[2][]%
192 {%
```

Zunächst setzen die die Optionen für das Paket `background`. Wir starten eine Gruppe um die Wirkung der Optionen lokal zu halten und setzen abschließend die verschiedenen Formularabschnitte. Falls ein `cc` (Carbon copy/Kopie) gewünscht wird, setzen wir den Dokumentbody erneut auf einer neuen Seite und setzen auch das entsprechende Wasserzeichen im Hintergrund.

```

193 \centering%
194 \backgroundsetup%
195 {%
196   contents=\MLG@JK@bgopt@contents,%
197   color=\MLG@JK@bgopt@color,%
198   opacity=\MLG@JK@bgopt@opacity,%
199   scale=\MLG@JK@bgopt@scale%
200 }%
201 \begingroup%
202   \setkeys{milog}{#1}%
203   \MLG@JK@colortbl@transparent%
204   \MLG@JK@doc@body{#2}%
205   \ifthenelse{\equal{\MLG@JK@cc}{true}}%
206   {%
207     \newpage%
208     \BgThispage%
209     \MLG@JK@doc@body{#2}%
210   }%
211   {}%
212 \endgroup%
213 }%

214 </class>

215 <{*milogsh>

```

7.2 milog

Das folgende `milog` Bash-Script zeigt beispielhaft eine mögliche Implementierung zur automatischen Auswertung eines CSV-Exports aus einer Arbeitszeiterfassungs-App am Beispiel „Arbeitszeiterfassung“ [2].

```

216 #!/bin/bash
217 #
218 VERSION="v1.0 (C) Josef Kleber (LPPL)"

```

Zunächst definieren wir einige Defaults, insbesondere für die CSV- und PDF-Ausgabedatei, sowie „boolsche“ Schalter für die „CSV“ und „LATEX“ Durchgänge.

```

219 INPUT=""
220 OUTPUT="_milog_.csv"
221 DATE=""
222 PDF="milog"
223 CSV="true"
224 LATEX="true"
225 #

```

Nun können wir die Kommandozeilen-Argumente auswerten:

-i CSV-Eingabedatei

- o CSV-Ausgabedatei
- p PDF-Ausgabedatei
- d Datum/Zeitspanne, üblicherweise im Format MM/YYYY
- c CSV only → LATEX="false"
- l LATEX only

```

226 while getopts "i:o:d:p:cl" flag
227 do
228   case "$flag" in
229     i) INPUT="$OPTARG";;
230     o) OUTPUT="$OPTARG";;
231     p) PDF="$OPTARG";;
232     d) DATE="$OPTARG";;
233     c) LATEX="false";;
234     l) CSV="false";;
235   esac
236 done
237 #

```

Falls kein Datum übergeben wurde, versuchen wir den Namen der CSV-Eingabedatei im Format YYYYMM auszuwerten.

```

238 if [ "$DATE" = "" ]
239 then

```

Zunächst definieren wir eine „regular expression“ für eine sechsstellige Zahl und besorgen uns den Namen der CSV-Eingabedatei ohne Dateieindung.

```

240 regex='^[0-9]{6}$'
241 YEARMONTH='echo "$INPUT" | cut -d'.' -f1'

```

Gelingt ein „Match“, belegen wir DATE entsprechend im Format MM/YYYY.

```

242 if [[ "$YEARMONTH" =~ $regex ]]
243 then
244   YEAR='echo "$YEARMONTH" | cut -c1-4'
245   MONTH='echo "$YEARMONTH" | cut -c5-6'
246   DATE="$MONTH/$YEAR"
247 fi
248 fi
249 #

```

Im CSV-Durchgang lesen wir zunächst die CSV-Eingabedatei ein, transformieren sie in das von milog.cls benötigte Format und schreiben sie in die CSV-Ausgabedatei.

```

250 if [ "$CSV" = "true" ]
251 then
252   echo "reading $INPUT"
253   cat "$INPUT" | \

```

```

254 # delete first 3 lines
255 head -n -4 | \
256 # delete last 3 lines
257 tail -n +4 | \
258 # deal with "empty" lines
259 sed -e 's/;;;;;;;;;$/{};{};{};{};{};{};{};{};{};{}/' | \
260 # delete some columns we don't need
261 cut -d ';' -f 2-4,8-10 | \
262 # replace the last ';' with ';'
263 sed -e 's/;$;/{}/' | \
264 # replace header keys
265 sed -e 's/Datum/tag/' -e 's/von/beginn/' -e 's/bis/ende/' -e 's/Pausen/pause/' -e 's/Net
266 # change 'Ruhetag' lines
267 sed -e 's/\(.*\);Ruhetag;Ruhetag;.*\1;{};{};{};{};{};{};{};{};{};{}/' | \
268 # change 'Ferien/Urlaub' lines
269 sed -e 's/\(.*\);Ferien/Urlaub;Ferien/Urlaub;.*\1;{};{};{};{};{};{};{};{};{};{}/' | \
270 # change 'Krank' lines
271 sed -e 's/\(.*\);Krank;Krank;.*\1;{};{};{};{};{};{};{};{};{};{}/' | \
272 # change 'Feiertag' lines
273 sed -e 's/\(.*\);Feiertag;Feiertag;.*\1;{};{};{};{};{};{};{};{};{};{}/' | \
274 # change last line with total hours
275 sed -e 's/^;;; \(.*\) /nan;nan;nan;nan;\1;nan/' | \
276 # replace '0:00' with '{}'
277 sed -e 's/0:00{/g' | \
278 # change order of columns and field separator ';' -> '\t'
279 awk 'BEGIN { OFS="\t"; FS=";" } { print $1, $2, $4, $3, $5, $6 }' >"$OUTPUT"
280 echo "writing $OUTPUT"
281 fi
282 #

```

Im LATEX-Durchgang prüfen wir zuerst auf „LATEX only“ und passen die CSV-Ausgabedatei entsprechend an.

```

283 if [ "$LATEX" = "true" ]
284 then
285   if [ "$CSV" = "false" ]
286   then
287     if [ -z "$INPUT" ]
288     then
289       OUTPUT="_milog_.csv"
290     else
291       OUTPUT="$INPUT"
292     fi
293   fi

```

Wir definieren eine HERE-L^AT_EX-Datei ...

```

294 cat >"$PDF.tex" <<EOF
295 \documentclass{milog}
296 \usepackage[utf8]{inputenc}
297 \usepackage[T1]{fontenc}
298 \usepackage{libertine}

```

```
299 \begin{document}
300 \milog[month={\DATE}]{\OUTPUT}
301 \end{document}
302 EOF
303 echo "creating $PDF.pdf"
```

und erzeugen damit unsere PDF-Datei.

```
304 pdflatex "$PDF" 2>&1 >/dev/null
305 rm "$PDF".tex "$PDF".aux "$PDF".log
306 fi
307 #
308 exit 0

309 \</milogsh>
```


8 Literatur

- [1] Hendri Adriaens. The xkeyval package, 2014.
<http://mirrors.ctan.org/macros/latex/contrib/xkeyval/xkeyval.pdf>.
- [2] askanimus. Arbeitszeiterfassung, 2016.
<https://play.google.com/store/apps/details?id=askanimus.arbeitszeiterfassung>.
- [3] Bundesministerium der Justiz und für Verbraucherschutz (in Zusammenarbeit mit der juris GmbH). Gesetz zur Regelung eines allgemeinen Mindestlohns (Mindestlohngesetz - MiLoG), 2014.
<https://www.gesetze-im-internet.de/bundesrecht/milog/gesamt.pdf>.
- [4] David Carlisle. The colortbl package, 2012.
<http://mirrors.ctan.org/macros/latex/contrib/colortbl/colortbl.pdf>.
- [5] David Carlisle. The tabularx package, 2014.
<http://mirrors.ctan.org/macros/latex/required/tools/tabularx.pdf>.
- [6] Simon Fear. Publication quality tables in L^AT_EX, 2005.
<http://mirrors.ctan.org/macros/latex/contrib/booktabs/booktabs.pdf>.
- [7] Dr. Christian Feuersänger. Manual for Package pgfplotstable, 2016.
<http://mirrors.ctan.org/graphics/pgf/contrib/pgfplots/doc/pgfplotstable.pdf>.
- [8] Gonzalo Medina. The background package, 2014.
<http://mirrors.ctan.org/macros/latex/contrib/background/background.pdf>.
- [9] Josselin Noirel. The xifthen package, 2015.
<http://mirrors.ctan.org/macros/latex/contrib/xifthen/xifthen.pdf>.
- [10] Heiko Oberdiek. The transparent package, 2016.
<http://mirrors.ctan.org/macros/latex/contrib/oberdiek/transparent.pdf>.
- [11] Hideo Umeki. The geometry package, 2010.
<http://mirrors.ctan.org/macros/latex/contrib/geometry/geometry.pdf>.

9 Change History

v0.1		made id optional in idtab	11
	General: initial version	7	
v0.2		v0.5	
	General: added \milogsetup . . .	12	General: added option cc
	added loading of milog.cfg . . .	11	changed color default to
v0.3			gray40
	General: added milog bash script	13	made „colortbl“ transparent
v0.4			10
	General: added \Formular	10	v1.0
			General: IPO CTAN
			7

Symbols

- \@plus 99, 101
 - \@tempdima 93, 94, 95, 99
 - \@tempdimb 94, 96
 - \@tempdimc 95, 100
- A**
- \advance 94, 95
 - \arraystretch 149, 162
 - article.cls (Class) 4, 8
 - \AtBeginDocument 139
- B**
- background (Package) 12
 - \backgroundsetup 194
 - \bfseries 46, 150
 - \BgThispage 208
 - \bottomrule 86
- C**
- \centering 170, 193
 - Class
 - article.cls 4, 8
 - milog.cls 3, 4, 6, 7, 14
 - colortbl (Package) 10
 - \CT@do@color 90
 - \CT@do@color 92
 - \CurrentOption 30
- D**
- \DeclareOptionX 30
 - \define@key 9, 13
 - \documentclass 295
 - \dotfill 169
- F**
- \footnotesize 170
 - \Formular 104
- G**
- \global 92
- H**
- \hskip 99, 101
 - \hspace 177
- I**
- \IfFileExists 141
 - \input 143
- K**
- \kern 96, 100
- L**
- \Large 45
 - \leaders 98
 - \let 92
 - \LoadClass 34
- M**
- \midrule 84
 - \milog 191, 300
 - milog.cls (Class) .. 3, 4, 6, 7, 14
 - \milogsetup 187
 - \MLG@JK@bgopt@color .. 55, 197
 - \MLG@JK@bgopt@contents 54, 196
 - \MLG@JK@bgopt@opacity 56, 198
 - \MLG@JK@bgopt@scale .. 57, 199
 - \MLG@JK@cc 205
 - \MLG@JK@color 85
 - \MLG@JK@colortbl@transparent
 - 88, 203
 - \MLG@JK@company 151
 - \MLG@JK@companyname 151
 - \MLG@JK@csvgtab 159, 184
 - \MLG@JK@csvgtab@as 52, 162
 - \MLG@JK@csvgtab@opacity 53, 97
 - \MLG@JK@csvgtab@topskip 48, 161
 - \MLG@JK@define@key 6, 18, 19, 20,
 - 21, 22, 23, 24, 25, 26, 27, 29
 - \MLG@JK@doc@body 180, 204, 209
 - \MLG@JK@heading 182
 - \MLG@JK@heading@font@size 45,
 - 182
 - \MLG@JK@heading@font@style 46,
 - 182
 - \MLG@JK@id 153, 155
 - \MLG@JK@idname 155
 - \MLG@JK@idtab 146, 183
 - \MLG@JK@idtab@as 51, 149
 - \MLG@JK@idtab@topskip 47, 148
 - \MLG@JK@idtab@width .. 50, 150
 - \MLG@JK@month 156
 - \MLG@JK@monthname 156
 - \MLG@JK@name 152
 - \MLG@JK@namename 152
 - \MLG@JK@signature 165, 176, 178

\MLG@JK@signature@employee	61, 176	\vrule	98
\MLG@JK@signature@employer	62, 178	W	
\MLG@JK@signature@labelskip	60, 169	\wd	93, 101
\MLG@JK@signature@width	59, 167	X	
\MLG@JK@signaturebox	173, 185	xkeyval (Package)	7
\MLG@JK@signaturebox@space	58, 177	xkvLxp (Package)	7
\MLG@JK@signaturebox@topskip	49, 175	Z	
		\z@	93, 101
N			
\newpage	207		
P			
Package			
background	12		
colortbl	10		
xkeyval	7		
xkvLxp	7		
\pagestyle	44		
\parbox	167		
\parindent	43		
\PassOptionsToClass	30		
\pgfplotstableread	104		
\pgfplotstableset	63		
\pgfplotstabletypeset	163		
\ProvidesClass	3		
R			
\rowcolor	85		
S			
\S	28		
\setkeys	189, 202		
\setlength	43		
T			
\toprule	84		
\transparent	97		
U			
\usepackage	296, 297, 298		
V			
\vfill	47, 48, 49		