The isodateo package*

Harald Harders
h.harders@tu-bs.de

File Date 2000-08-08, Printed 2001-05-17

Abstract

This package provides commands to switch between different date formats (standard, ISO, numeric, LaTeX package). They are used by the \today command and by the \printdate and \printdateTeX commands that print any date. This package supports German (old and new rules, Austrian), US English, and all languages that have the same date format as British English does[1].

The idea for this package was taken from the akletter class.

Contents

1 Commands

1.1 Switching the date format

\today This package provides five commands to switch the output format of the \today, the \printdate, and the \printdateTeX commands:

*This file has version 1.06 last revised 2000-08-08, documentation dated 2000-08-08.

[1]E.g. Danish, French

`\isodate`	date format described in ISO 8601 and DIN 5008 (yyyy-mm-dd)
`\numdate`	numeric date format with four digits of the year
`\shortdate`	short numeric date format with two digits of the year
`\TeXdate`	date format used for version description of packages (yyyy/mm/dd)
`\origdate`	original LaTeX format

`\isodate` `\numdate` `\shortdate` `\TeXdate` `\origdate`

The numeric and short numeric format change their behaviour depending on the actual language:

German, nGerman	`dd.\,mm.~yyyy`	resp.	`dd.\,mm.\,yy`
US English	`mm/dd/yyyy`	resp.	`mm/dd/yy`
other languages	`dd/mm/yyyy`	resp.	`dd/mm/yy`

So this package supports German (old and new rules, Austrian), US English, and all languages that have the same date format as British English does[2]. Switching the language by using \selectlanguage also switches back to the original date format.

1.2 Printing any date

`\printdate` The command `\printdate{#1}` prints any date in the actual format. The argument may be a date in German, British English, or ISO format, e.g.

```
\printdate{24.12.2000}
\printdate{24/12/2000}
\printdate{2000-12-24}
```

`\printdateTeX` The command `\printdateTeX{#1}` prints any date in the actual format. The argument must be in the LaTeX format `yyyy/mm/dd`, e.g.

```
\printdateTeX{2000/12/24}
```

This command is useful for printing version information stored in a macro. For example the version of this documentation is stored in the macro `\docdate` ("2000/08/08"). To print it with the actual date format you can use the command `\printdateTeX{\docdate}` which leads to "2000-08-08".

1.3 Changing the ISO format

`\isodash` I am not sure whether the ISO format should be yyyy-mm-dd or yyyy–mm–dd. By default I use "-" as dash. You can change this using the `\isodash` command, e.g.

```
\printdate{24/12/2000}
\isodash{--}
\printdate{24/12/2000}
```

leads to "2000-12-24 2000–12–24". Or for example

```
\isodash{$\cdot$}
\printdate{24/12/2000}
```

leads to "2000·12·24".

[2] E.g. Danish, French

2 Calling the package

The package is called using the `\usepackage` command:
`\usepackage[option]{isodate}`.

The possible package options can be seen in table 1.

Table 1: Package options

option	used date format
`iso`	ISO date format
`num`	numeric date format with 4 digits of the year
`short`	numeric date format with 2 digits of the year
`TeX`	LaTeX numeric date format (yyyy/mm/dd)
`orig`	normal LaTeX date format (default)

A Licence

B Known errors

- The `\printdate` and `\printdateTeX` commands are not very good in checking the argument for correct syntax.
- For the language American: Using the package `babel` only the language name "american" works, using the package `*german` only "USenglish" works.

C Planned features and changes

- Of course eliminate the errors.
- Add other languages then german, ngerman, english, USenglish. Please help me with this topic, I don't know the date formats in other languages.

D The implementation

Heading of the package:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{isodateo}[\filedate]
\RequirePackage{ifthen}
\RequirePackage{calc}
\IfFileExists{substr.sty}{\RequirePackage{substr}%
 }{\PackageError{isodateo.sty}{Package file substr.sty not found}
   {This version of isodateo.sty needs the package substr.sty.^^J%
     You can download it from CTAN:/macros/latex/contrib/supported/substr/^^J%
```

```
    E.g. one CTAN node is ftp.dante.de.
    Install substr.sty into your TeX tree.}}
```

At the end of the preamble the package tests whether one of the packages `babel`, `german`, or `ngerman` is loaded. If not it is assumed that American English is wanted (LaTeX is an American programme). The original date format is saved and the command `\iflanguage` is redefined to process the "true part" for english, american, and USenglish options and otherwise the "false part".

```
\AtBeginDocument{%
  \@ifpackageloaded{babel}{}{%
    \@ifpackageloaded{german}{}{%
      \@ifpackageloaded{ngerman}{}{%
```

Here you can add new languages. Tell me what you have inserted in order to enable me to actualize the package.

```
        \let\dateamerican\today%
        \setboolean{isodate@american}{true}%
        \def\iflanguage#1#2#3{\ifthenelse{%
            \equal{#1}{english}\or%
            \equal{#1}{american}\or%
            \equal{#1}{USenglish}%
            }{#2}{#3}}%
        }}}}
```

Declare the boolean variable isodate@american. This is necessary because the command \iflanguage cannot decide if the language is English or American.

```
\newboolean{isodate@american}%
% \changes{1.06}{2000/08/08}{Avoid using the hack with redefining
% \selectlanguage}
% Define the package options.
%    \begin{macrocode}
\DeclareOption{iso}{\AtBeginDocument{\isodate}}
\DeclareOption{num}{\AtBeginDocument{\numdate}}
\DeclareOption{short}{\AtBeginDocument{\shortdate}}
\DeclareOption{TeX}{\AtBeginDocument{\TeXdate}}
\DeclareOption{orig}{\AtBeginDocument{\origdate}}
\ExecuteOptions{orig}
\ProcessOptions
```

Print day or month filled with zero to a format with two digits.

```
\def\dday{\ifthenelse{\number\day<10}{0}{}\number\day}
\def\dmonth{\ifthenelse{\number\month<10}{0}{}\number\month}
```

Print day and month in numerical format using the right format for the present language.

```
\DeclareRobustCommand*{\num@today}[1]{%
  \iflanguage{german}{\dday.\,\dmonth.#1}{%
    \iflanguage{austrian}{\dday.\,\dmonth.#1}{%
      \iflanguage{ngerman}{\dday.\,\dmonth.#1}{%
        \iflanguage{naustrian}{\dday.\,\dmonth.#1}{%
```

Here you can add new languages. Tell me what you have inserted in order to enable me to actualize the package.

```
          \iflanguage{english}{%
            \ifthenelse{\boolean{isodate@american}}{%
              \dmonth/\dday/}{\dday/\dmonth/}}{%
```

```
  \dday/\dmonth/}}}}}%
}
```

\numdate

Switch to long numeric date format.

```
\DeclareRobustCommand*{\numdate}{%
```

Find out whether the language may be English or American. The English original date format does not contain a komma while the american does.

```
  \origdate%
  \setboolean{isodate@american}{false}%
  \iflanguage{american}{\IfCharInString{,}{\today}{%
    \setboolean{isodate@american}{true}}{}}{}%
```

Define the new \today command.

```
  \gdef\today{%
    \num@today{~}%
    \number\year}}
```

\shortdate

Switch to short numeric date format.

```
\newcounter{yeartwo}
\DeclareRobustCommand*{\shortdate}{%
```

Find out whether the language may be English or American. The English original date format does not contain a komma while the american does.

```
  \origdate%
  \setboolean{isodate@american}{false}%
  \iflanguage{american}{\IfCharInString{,}{\today}{%
    \setboolean{isodate@american}{true}}{}}{}%
```

Define the new \today command.

```
  \gdef\today{%
    \num@today{\,}%
    \setcounter{yeartwo}{\number\year}%
    \whiledo{\theyeartwo>99}{\setcounter{yeartwo}{\theyeartwo-100}}{}%
    \ifthenelse{\number\theyeartwo<10}{0}{}\theyeartwo}}
```

\isodate

Switch to ISO date format.

```
\DeclareRobustCommand*{\isodate}{%
  \gdef\today{%
    \number\year\iso@isodash%
    \ifthenelse{\number\month<10}{0}{}\number\month\iso@isodash%
    \ifthenelse{\number\day<10}{0}{}\number\day}}
```

Define the default ISO dash to “-”.

```
\def\iso@isodash{-}%
```

\isodate

Define the command `\isodash` which changes the dash in the ISO date format.

```
\DeclareRobustCommand*{\isodash}[1]{\def\iso@isodash{#1}}%
```

\origdate

Switch back to original date format.

```
%\DeclareRobustCommand*{\origdate}{\gdef\today{\iso@origdate}}
\DeclareRobustCommand*{\origdate}{\csname date\languagename\endcsname}
```

`\TeXdate` Switch to the TeX date format.

```
\DeclareRobustCommand*{\TeXdate}{%
  \gdef\today{%
    \number\year/%
    \ifthenelse{\number\month<10}{0}{}\number\month/%
    \ifthenelse{\number\day<10}{0}{}\number\day}}
```

Print any date (internal command, syntax: `\iso@printdate{yyyy}{mm}{dd}`).

```
\DeclareRobustCommand*{\iso@printdate}[3]{%
  \begingroup%
  \def\year{#1}%
  \def\month{#2}%
  \def\day{#3}%
  \today%
  \endgroup%
}
```

Define counters to count the numbers of special characters in the arguments of the `\printdate` and `\printdateTeX` commands.

```
\newcounter{iso@slash}
\newcounter{iso@minus}
\newcounter{iso@dot}
```

`\printdate` Print any date in the actual date format. This command understands the German, British, and ISO formats.

```
\DeclareRobustCommand*{\printdate}[1]{%
  \expandafter\iso@expafterprintdate\expandafter{#1}}%
```

`\iso@expafterprintdate` The command `\iso@expafterprintdate` needs an already expanded argument. So the command `\printdate` expands it and calls `\iso@expafterprintdate`.

The error handling of this macro is very poor. It is just tested if either a "/", "-", or "." is included in the argument twice. It is not tested if the argument consists of numbers, only.

```
\DeclareRobustCommand*{\iso@expafterprintdate}[1]{%
  \SubStringsToCounter{iso@slash}{/}{#1}%
  \SubStringsToCounter{iso@minus}{-}{#1}%
  \SubStringsToCounter{iso@dot}{.}{#1}%
  \ifthenelse{\equal{\theiso@dot}{2}}{\printdatenumger{#1}}{%
    \ifthenelse{\equal{\theiso@minus}{2}}{\printdateiso{#1}}{%
      \ifthenelse{\equal{\theiso@slash}{2}}{\printdatenumeng{#1}}{%
        ????\iso@isodash ??\iso@isodash ??%
        \PackageError{isodateo}{unrecognized date format}{Use one of
          the following formats as macro argument:^^J%
          \space\space dd.mm.yyyy^^J%
          \space\space dd/mm/yyyy^^J%
          \space\space yyyy-mm-dd^^J%
          Don't use any spaces or commands like \protect\, or
          \protect~ inside the argument.}%
        }}}}
```

Analyze the argument containing a date in ISO format an print it. This macros does not contain any error handling.

```
\DeclareRobustCommand*{\printdateiso}[1]{%
```

```
  \expandafter\iso@printdateiso #1\@empty}
\def\iso@printdateiso#1-#2-#3\@empty{\iso@printdate{#1}{#2}{#3}}
```

Analyze the argument containing a date in German numeric format an print it. This macros does not contain any error handling.

```
\DeclareRobustCommand*{\printdatenumger}[1]{%
  \expandafter\iso@printdatenumger #1\@empty}
\def\iso@printdatenumger#1.#2.#3\@empty{\iso@printdate{#3}{#2}{#1}}
```

Analyze the argument containing a date in Britisch English numeric format an print it. This macros does not contain any error handling.

```
\DeclareRobustCommand*{\printdatenumeng}[1]{%
  \expandafter\iso@printdatenumeng #1\@empty}
\def\iso@printdatenumeng#1/#2/#3\@empty{\iso@printdate{#3}{#2}{#1}}
```

`\printdateTeX` Analyze the argument containing a date in the LaTeX style yyyy/mm/dd an print it. This format can not be handled automaticaly by `\printdate` because it could be mixed up with the English format. The error handling of this routine is very poor. It just checks whether the argument contains at least one "/".

```
\DeclareRobustCommand*{\printdateTeX}[1]{%
  \expandafter\iso@printdateTeX\expandafter{#1}}
\DeclareRobustCommand*{\iso@printdateTeX}[1]{%
  \SubStringsToCounter{iso@slash}{/}{#1}%
  \ifthenelse{\equal{\theiso@slash}{2}}%
  {\expandafter\iso@@printdateTeX #1\@empty}{%
    ????\iso@isodash ??\iso@isodash ??%
    \PackageError{isodateo}{unrecognized date format}{Use the format
      yyyy/mm/dd.^^J%
      Don't use any spaces or commands like \protect\, or
      \protect~ inside the argument.}}%
}
\def\iso@@printdateTeX#1/#2/#3\@empty{\iso@printdate{#1}{#2}{#3}}
```

The end of the package.

Change History

1.01

General: Improve documentation . 1

1.02

General: Fix American language support by a hack 4

1.03

General: Insert code for handling not loaded language packages . 4

1.04

General: Add LaTeX date format yyyy/mm/dd 7

Make the commands robust . . . 1

`\TeXdate`: Add LaTeX date format yyyy/mm/dd 6

1.05

General: Change all internal command names to start with \iso@ 1

Note that every language that has the same format as English is supported. 2

Throw out the commands \IfSubStringInString and \IfCharInString and use the package substr.sty instead 1

`\iso@expafterprintdate`: Count appearances of "/", ".", and "-" and complain if not at least one of them is equal to 2 6

\printdateTeX: Count appearances of "/" and complain if not equal to 2 7

1.06

\numdate: Choose between English and American language 5

\origdate: Use the command \date*language* to switch back to the original date format . . . 5

\shortdate: Choose between English and American language . . 5

Index

Numbers written in italic refer to the page where the corresponding entry is described, the ones underlined to the code line of the definition, the rest to the code lines where the entry is used.