The bxwareki package

Takayuki YATO　(aka. “ZR”)

v0.7　[2023/07/07]

1 Overview

This package provides commands to convert from the Gregorian calendar (2022/8/28) to the Japanese rendering of the Japanese calendar (令和 4 年 8 月 28 日). You can choose whether the numbers are written in Western numerals (like 28) or kanji numerals (*kansuji*, like 二八).

Note that the package only deals with dates in the year 1873 or later, where the Japanese calendar (*wareki*; 和暦) can be regarded as a variant of Gregorian calendar with the different notation of years. *Note:* To avoid confusion, this document refers to the original Gregorian calendar as the *Western calendar*, which corresponds to the Japanese term *seireki* (西暦).

System requirement

- TEX format: LATEX.
- TEX engine: pdfTEX, LuaTEX, XETEX, pTEX, upTEX, ApTEX (pTEX-ng), NTT-JTEX.

This package is designed to be very generic, and it could work on engines other than those mentioned above. Starting from v0.7, when the package *knows* that it cannot properly handle kanji characters on the engine in use, then it will switch to the “fallback mode”, where all kanji characters are replaced with a ‘?’ symbol. Users can use `\WarekiIfAvailable` to know whether the package really works. It is expected that this package never fails on load.

2 Package Loading

No options are available.

```
\usepackage{bxwareki}
```

3 Usage

3.1 Conversion from the given date

- `\warekisetdate{`⟨*year*⟩`}{`⟨*month*⟩`}{`⟨*day*⟩`}` : Converts from the specified Western date. This command itself prints nothing and the result will be rendered by the commands described at the following items, where the result for the invocation `\warekisetdate{2022}{8}{28}` will be shown as example.

  Note: Although the Japanese calendar differs from the Western calendar only in the notation of years, the value of months and days are still required, since the notation of the year in which *kaigen* (改元; change of gengo) occurs depends on months and days.

- `\warekisettoday` : Does `\warekisetdate` with the current date.

- Counter `warekiyear` : The year number (within the gengo); e. g. “4”.

  Note: Unlike ordinary counters, the assignment to `warekiyear` by `\warekisetdate` is *local*. Moreover, manual assignment to this counter is not supported.

- `\warekigengo` : The gengo (元号) in kanji, e. g. “令和”.

- `\warekigengoinitial` : The initial Latin letter of the gengo, e. g. “R”.

- `\warekiyear` : The full notation of the year (without ‘年’), e. g. “令和 4”.

  Note: When the year number is one, the kanji ‘元’ is used instead of the numeral ‘1’.

- `\warekidate` : The date string, e. g. “令和 4 年 8 月 28 日”.

- `\warekikanjidate` : The date string using kansuji (kanji numerals), e. g. “令和四年八月二八日”.

- `\warekijkanjidate` : The date string using “kansuji-by-reading” (that is, kanji rendering of numbers in the Japanese language), e. g..“令和四年八月二十八日”.

- `\warekicustomdate{`⟨*option*⟩`}` : The date string in the form specified with the option. The option is a string of letters such as `wk` and each letter means a specific setting. When the option is empty, the date is rendered in the form “2022 年 8 月 28 日” (using the Western calendar). The available option letters are:

  - `w` : uses Japanese calendar (2022 年 → 令和 4 年);
  - `k` : uses kansuji (28 → 二八);
  - `j` : uses kansuji-by-reading (28 → 二十八);

    Note: Western years does not support kansuji-by-reading and thus `k` will be applied instead (二〇一八, not 二千十八).

  - `J` : variant of `j` where “ten’s multiple” kanji characters (廿 and 卅) are employed (28 → 廿八);
  - `o` : uses *imyo* (異名) for months[1] (8 月 → 葉月).

  Note: This command is supported only on pdfLaTeX, XeLaTeX, LuaLaTeX, upLaTeX, ApLaTeX and recent versions of pLaTeX. On other engines it simply falls back to `\warekidate`.

- `\WarekiIfCustomDateAvailable{`⟨*true*⟩`}{`⟨*false*⟩`}` : Tests if the command `\warekicustomdate` is supported on the engine in use.

3.2 Current date

There are also handy commands solely for printing the current date. These commands always represent the current date, and are not affected by `\warekisetdate` or `\warekisettoday`.

- `\warekitoday` : The `\warekidate` form of the current date.

- `\warekikanjitoday` : The `\warekikanjidate` form.

- `\warekijkanjitoday` : The `\warekijkanjidate` form.

[1] Don’t ask me if this form is ever used in LaTeX document!

3.3 Inter-alphabet-kanji spaces

In quality Japanese typesetting, a thin space (*shibuaki*; 四分空き) must be inserted between an alphabet letter and a kanji letter. This package by default inserts a command suitable for the most prevalent Japanese-typesetting environment for the engine in use.

- On pLaTeX, upLaTeX and ApLaTeX: Nothing is inserted, since the engine can automatically insert shibuaki spaces.
- On LuaLaTeX and XeLaTeX: Nothing is inserted, on the assumption that the package for Japanese typesetting (such as LuaTeX-ja and xeCJK) will automatically insert shibuaki spaces.
- On LaTeX and pdfLaTeX: A tie (`~`) is inserted, on the assumption that the CJK package is employed and `\CJKtilde` is in effect.

The command for shibuaki can be changed with the following commands:

- `\WarekiUseNormalInterGlue` : Uses the normal setting, as mentioned above.
- `\WarekiUseNoInterGlue` : Disables shibuaki spaces.
- `\WarekiUseCustomInterGlue{`⟨*text*⟩`}` : Uses ⟨*text*⟩ for making shibuaki spaces.

  Note: This command is supported only on engines with ε-TeX extension.

3.4 Counter output commands

The following commands are intended to use with `warekiyear` counter, but they can probably be used as general-use counter output commands (like `\arabic`):

- `\WarekiKansuji{`⟨*counter*⟩`}` : Prints the counter value using kansuji.
- `\WarekiJKansuji{`⟨*counter*⟩`}` : Prints the counter value using kansuji-by-reading. Only valid for numbers less than 1000.

4 Notices for TeX programmers

4.1 Expandability of the commands

- On the engines with native kanji/Unicode support (i. e. LuaLaTeX, XeLaTeX, pLaTeX, upLaTeX, and ApLaTeX), the content (one-level expansion) of `\wareki...date` (except `\warekicustomdate`) and `\wareki...today` is a simple string of character tokens, unless `\WarekiUseCustomInterGlue` is in effect. The same holds for LaTeX and pdfLaTeX, except that each kanji character is represented by the sequence of activated byte tokens and `~` is inserted as shibuaki spaces.
- On the engines with native kanji/Unicode support, `\warekicustomdate` fully expands to a simple string of character tokens (again without `\WarekiUseCustomInterGlue`), and the situation on LaTeX and pdfLaTeX is parallel to that described at the previous item.
- When `\WarekiUseCustomInterGlue` is used with some argument, the content of `\wareki...date` and `\wareki...today` could contain some occurrences of the argument. If the argument is fully expandable, the commands are still fully expandable on the engines with native kanji/Unicode support.