

The `bxwareki` package

Takayuki YATO (aka. “ZR”)

v0.6 [2019/06/01]

1 Overview

This package provides commands to convert from the Gregorian calendar (2018/8/28) to the Japanese rendering of the Japanese calendar (平成 30 年 8 月 28 日). You can choose whether the numbers are written in Western numerals (28) or kanji numerals (二八). The package is aware of the current era name, *Reiwa* (令和).

Note that the package only deals with dates in the year 1873 or later, where the Japanese calendar is really a Gregorian calendar with the different notation of years.

System requirement

- T_EX format: L^AT_EX.
- T_EX engine: pdfT_EX, LuaT_EX, X_YT_EX, pT_EX, upT_EX, ApT_EX (pT_EX-ng), NTT-jT_EX.

Note: The modern (post-1873) Japanese calendar (*wareki*; 和暦) can be regarded as a variant of Gregorian calendar where only the notation of years differs from the original. To avoid confusion, this document refers to the original Gregorian calendar as the *Western calendar*, which corresponds to the Japanese term *seireki* (西暦).

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. The result will be rendered by the commands described at the following items, where the result for the invocation `\warekisetdate{2018}{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. “30”.

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

- `\warekigengo` : The gengo in kanji, e.g. “平成”.
 - `\warekigengoinitial` : The initial Latin letter of the gengo, e.g. “H”.
 - `\warekiyear` : The full notation of the year (without ‘年’), e.g. “平成 30”.
Note: When the year number is one, the kanji ‘元’ is used instead of the numeral ‘1’.
 - `\warekideate` : The date string, e.g. “平成 30 年 8 月 28 日”.
 - `\warekikanjideate` : The date string using kansuji (kanji numerals), e.g. “平成三〇年八月二八日”.
 - `\warekijkanjideate` : The date string using “kansuji-by-reading” (that is, kanji rendering of numbers in the Japanese language), e.g. “平成三十年八月二十八日”.
 - `\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 “2018 年 8 月 28 日” (using the Western calendar). The available option letters are:
 - `w` : uses Japanese calendar (2018 年 → 平成 30 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 months¹ (8 月 → 葉月).
- Note:* This command is supported only on pdfL^AT_EX, X_YL^AT_EX, LuaL^AT_EX, upL^AT_EX, ApL^AT_EX and recent versions of pL^AT_EX. On other engines it simply falls back to `\warekideate`.
- `\WarekiIfCustomDateAvailable{<true>}{<>false>}` : Tests if the command `\warekicustomdate` is supported on the engine in use.

3.2 Current date

These commands always represent the current date, and are not affected by `\warekisetdate` (or `\warekisettoday`).

- `\warekitoday` : The `\warekideate` form of the current date.
- `\warekikanjitoday` : The `\warekikanjideate` form.
- `\warekijkanjitoday` : The `\warekijkanjideate` form.

3.3 Inter-alphabet-kanji glues

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 pL^AT_EX, upL^AT_EX and ApL^AT_EX: Nothing is inserted, since the engine can automatically insert shibuaki spaces.

¹Don’t ask me if this form is ever used in L^AT_EX document!

- On Lua \LaTeX and Xe \LaTeX : Nothing is inserted, on the assumption that the package for Japanese typesetting (such as Lua \TeX -ja and xeCJK) will automatically insert shibuaki spaces.
- On \LaTeX and pdf \LaTeX : ~ is inserted, on the assumption that the CJK package is employed and $\backslash\text{CJKtilde}$ is in effect.

On the engines with ε - \TeX extension, the command for shibuaki can be changed with the following commands:

- $\backslash\text{WarekiUseCustomInterGlue}\langle text \rangle$: Uses $\langle text \rangle$ for making shibuaki spaces.
- $\backslash\text{WarekiUseNormalInterGlue}$: Reverts $\backslash\text{WarekiUseCustomInterGlue}$ and uses the normal setting.

4 Notices for \TeX programmers

- On the engines with native kanji/Unicode support (i. e. Lua \LaTeX , Xe \LaTeX , p \LaTeX , up \LaTeX , and Ap \LaTeX), the content (one-level expansion) of $\backslash\text{wareki}\dots\text{date}$ (except $\backslash\text{warekicustomdate}$) and $\backslash\text{wareki}\dots\text{today}$ is a simple string of character tokens, unless $\backslash\text{WarekiUseCustomInterGlue}$ is in effect. The same holds for \LaTeX and pdf \LaTeX , 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, $\backslash\text{warekicustomdate}$ fully expands to a simple string of character tokens (again without $\backslash\text{WarekiUseCustomInterGlue}$), and the situation on \LaTeX and pdf \LaTeX is parallel to that described at the previous item.
- The use of $\backslash\text{WarekiUseCustomInterGlue}$ does not break the full expandability of $\backslash\text{wareki}\dots\text{date}$ and $\backslash\text{wareki}\dots\text{today}$ on the engines with native kanji/Unicode support.