

# KKsymbols Package Documentation

Kosei Kawaguchi a.k.a. KKT<sub>E</sub>X

Version 2.3.0 (2026/08/20)

目次

1    Outline ..... 3

2    Comparison with `luatexja-otf` ..... 4

    2.1    Command-by-command examples ..... 5

    2.2    Practical selection guide ..... 5

3    Acknowledgements / Credit ..... 5

4    Installation ..... 6

    4.1    Dependencies ..... 6

    4.2    Loading and Options ..... 6

5    Caution ..... 9

6    Commands ..... 9

    6.1    The maru series ..... 9

7    Roman numerals ..... 11

    7.1    Recommended usage: non-starred commands and one group-wide phantom ... 12

    7.2    Corner-case test sheet ..... 14

8    Automatic alignment (reference box) ..... 21

    8.1    Overriding the reference ..... 21

9    Rotation ..... 22

    9.1    Commands ..... 22

    9.2    Vertical mode ..... 23

10   The seihou series ..... 23

11   The kakko series ..... 24

    11.1   Additional Description: `\ichimoji` and `\zenkakuhabafixer` ..... 25

12   License ..... 26

13   Version History ..... 26

# 1 Outline

**Japanese** このドキュメントでは、主要な仕様説明を日本語と英語の両方で記述します。日本語の説明で実用上の注意点を先に述べ、その直後に対応する英語説明を置きます。

このパッケージは、既存の OTF グリフだけに頼るのではなく、「現在のフォント」「任意の引数」「横書き・縦書き」に合わせて、丸数字や括弧付き文字などの囲み記号を構成するために作られています。

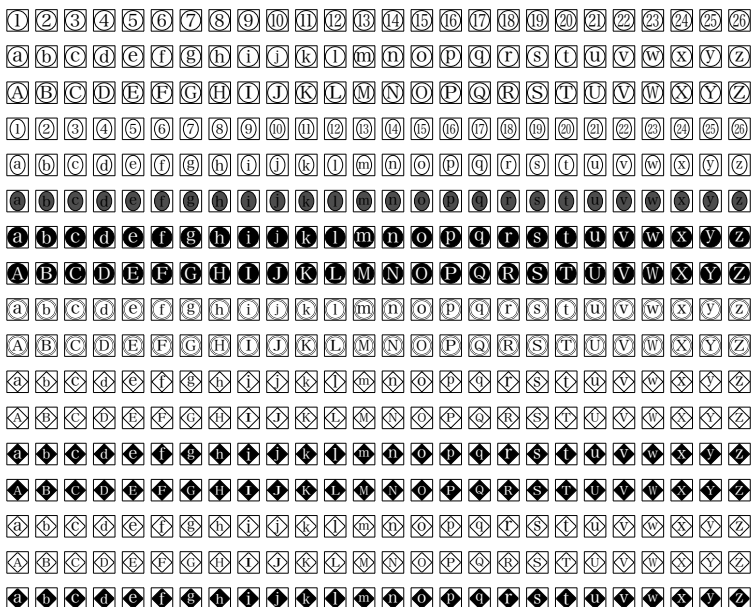
`luatexja-otf` の `\ajMaru` などは、Adobe-Japan1 の既存グリフを呼び出すため、品質が高く高速である一方、どの番号がどのグリフになるかを知るための早見表や CID 知識が必要になりがちです。本パッケージは図形と中身を組み合わせる再構成する設計なので、任意の文字列を同じインターフェースで囲めます。

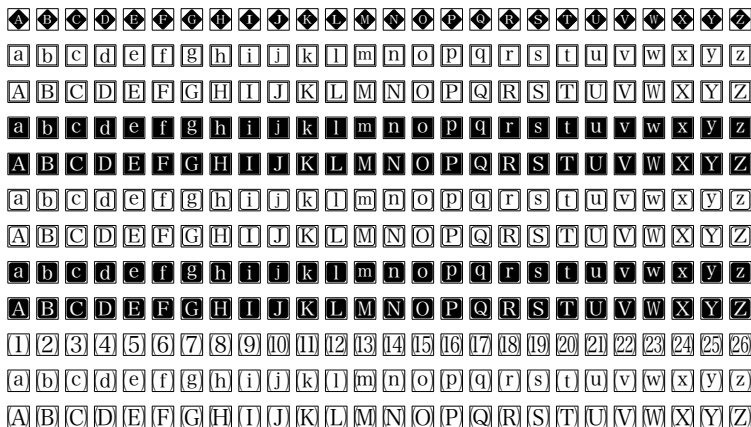
**English** This documentation now gives the important usage notes in both Japanese and English. The Japanese explanation states practical details first, followed by the corresponding English explanation.

This package is designed to build enclosed symbols such as circled numbers and parenthesized characters from the current font and arbitrary arguments, in both horizontal and vertical writing modes, rather than relying only on pre-existing OTF glyphs.

Commands such as `\ajMaru` in `luatexja-otf` call Adobe-Japan1 glyphs directly. This gives high-quality and fast output, but it often requires a reference chart or CID knowledge to know which number maps to which glyph. Kksymbols instead composes the enclosure and the content, so arbitrary strings can be enclosed through the same interface.

あ あ (あ) あ あ





## 2 Comparison with luatexja-otf

**Japanese** `luatexja-otf` は Adobe-Japan1 のグリフを直接呼び出すための強力なパッケージです。代表的なコマンドには、`\ajMaru`、`\ajKuroMaru`、`\ajKaku`、`\ajKakko`、`\ajRoman`、`\ajroman`、`\ajKakkoroman`、`\ajKakkoRoman` などがあります。これらは既存グリフをそのまま使うため、印刷品質は非常に高く、処理も軽量です。

一方で、`luatexja-otf` の囲み文字は「存在するグリフを選ぶ」方式です。たとえば `\ajMaru{1}` は丸数字のグリフを出しますが、`\ajMaru{ABC}` のような任意文字列を丸で囲む用途ではありません。対応範囲は Adobe-Japan1 側に存在するグリフと、`luatexja-ajmacros` が定義している番号範囲・合字名に依存します。

`KKsymbols` は反対に、「囲みの形」と「中身」をその場で組み合わせる方式です。そのため `\maru{ABC}`、`\kakko{QED.}`、`\seihou{長い文字列}` のように、数字以外の文字列や現在フォントの文字も同じ命令で処理できます。代償として、既存の完成グリフを呼び出す `luatexja-otf` よりも内部処理は重く、極端な引数では縮小や中央寄せの調整が見た目に出ることがあります。

**English** `luatexja-otf` is a powerful package for accessing Adobe-Japan1 glyphs directly. Typical commands include `\ajMaru`, `\ajKuroMaru`, `\ajKaku`, `\ajKakko`, `\ajRoman`, `\ajroman`, `\ajKakkoroman`, and `\ajKakkoRoman`. Since these commands use pre-designed glyphs, the output quality is excellent and the processing cost is low.

The important limitation is that `luatexja-otf` selects existing glyphs. For example, `\ajMaru{1}` outputs a circled-number glyph, but `\ajMaru{ABC}` is not a general command for enclosing an arbitrary string. Its coverage depends on the glyphs available in Adobe-Japan1 and on the numeric ranges and ligature names defined by `luatexja-ajmacros`.

`KKsymbols` takes the opposite approach: it composes the enclosure and the content at typesetting time. Therefore commands such as `\maru{ABC}`, `\kakko{QED.}`, and `\seihou{long text}` work with non-numeric strings and with the current font. The tradeoff is that `KKsymbols` does more internal work than `luatexja-otf`, and very long or unusual arguments may visibly depend on the package's

scaling and centering rules.

## 2.1 Command-by-command examples

**Japanese** 次の表は、同じような用途に見えるコマンドの比較です。左側の `luatexja-otf` は既存グリフ、右側の `KKsymbols` は現在フォントから組み立てた結果です。

**English** The following table compares commands that look similar in ordinary use. The `luatexja-otf` examples on the left are existing glyphs, while the `KKsymbols` examples on the right are composed from the current font.

表 1: Comparison with `luatexja-otf`

| Purpose              | <code>luatexja-otf</code>            | <code>KKsymbols</code>                 |
|----------------------|--------------------------------------|----------------------------------------|
| circled number       | <code>\ajMaru{8}</code> ⑧            | <code>\maru{8}</code> ⑧                |
| black circled number | <code>\ajKuroMaru{8}</code> ⑧        | <code>\kuromaru{8}</code> ⑧            |
| square number        | <code>\ajKaku{8}</code> ⑧            | <code>\seihou{8}</code> ⑧              |
| parenthesized number | <code>\ajKakko{8}</code> (8)         | <code>\kakko{8}</code> (8)             |
| uppercase roman      | <code>\ajRoman{8}</code> VIII        | <code>\kakko{\Rrnum*{8}}</code> (VIII) |
| lowercase roman      | <code>\ajroman{8}</code> viii        | <code>\kakko{\Rrnum{8}}</code> (viii)  |
| parenthesized roman  | <code>\ajKakkoroman{8}</code> (viii) | <code>\kakko{\Rrnum{8}}</code> (viii)  |

## 2.2 Practical selection guide

**Japanese** 完成済みの標準グリフが必要な場合、たとえば丸数字・括弧付き数字・ローマ数字が Adobe-Japan1 の範囲に収まる場合は、`luatexja-otf` が最も安定します。フォントベンダが設計した字形なので、紙面上の完成度も高くなります。

任意の文字列、現在の欧文・和文フォント、太字、色付き文字、長いラベル、独自の囲み形状が必要な場合は `KKsymbols` が向いています。`KKsymbols` は「グリフを探す」必要がなく、`\maru{任意}` や `\kakko{ABC}` のように直接書けます。

**English** Use `luatexja-otf` when you need standard pre-designed glyphs, such as circled numbers, parenthesized numbers, or roman numerals that are covered by Adobe-Japan1. Since these glyphs are designed by the font vendor, they usually give the most polished printed result.

Use `KKsymbols` when you need arbitrary strings, the current Latin or Japanese font, bold or colored content, long labels, or custom enclosure styles. `KKsymbols` does not require you to look up a glyph; you can write commands such as `\maru{任意}` or `\kakko{ABC}` directly.

## 3 Acknowledgements / Credit

In developing this package, I made extensive use of the advice I received from Mr. Yusuke Terada.

I recommend you to refer to his article when you develop new-type symbols on L<sup>A</sup>T<sub>E</sub>X.

<https://doratex.hatenablog.jp/entry/20211205/1638697391>

## 4 Installation

Place `KKsymbols.sty` in a directory where LaTeX can find it, e.g., your local `texmf` tree or alongside your document.

### 4.1 Dependencies

This package depends on the following packages.

- `LuaLaTeX-jā`
- `tikz`
- `clac`
- `luacode`
- `kvoptions`

### 4.2 Loading and Options

Load the package: `\usepackage[<options>]{KKsymbols}`

The following package options are provided.

**tsumesuji** To specify whether or not to “shrink” the box width occupied by “1”. By default, `tsumesuji=1` is specified, and the effect is activated. If you deactivate it, `tsumesuji=0` will do.

To clarify, this option is applied when the argument of a command subject to the `tsumesuji` option—after expansion—is a numeric string of two or more digits and contains the digit “1”. The difference is present in the following example:

| Input |                                     |
|-------|-------------------------------------|
| 1     | <code>% NOT APPLIED</code>          |
| 2     | <code>\kakko{\vphantom{X}12}</code> |
| 3     |                                     |
| 4     | <code>% APLIED</code>               |
| 5     | <code>\kakko{12}</code>             |

| Output |      |
|--------|------|
|        | (12) |
|        | (12) |

**autolowercase** Enables automatic lowercase sizing. The default is `autolowercase=1`. If the fully expanded argument is exactly one lowercase ASCII letter (`a--z`), a non-starred enclosure

command automatically uses the same sizing path as its starred form. This also recognizes one-letter results produced by `\Kran`, such as `\alphabetarray{2}` (which expands to `b`) and `\Rrnum{1}` (which expands to `i`). Multi-letter results such as `\Rrnum{8}` (`viii`), as well as digits, uppercase letters, Japanese text, and mixed strings, keep the normal sizing path. An explicit star remains available for multi-letter material.

`\jegg` is included in the same option. Its star continues to select the gray background; automatic lowercase sizing is applied independently to both `\jegg{...}` and `\jegg*{...}` while `autolowercase=1`. Set `autolowercase=0` to disable automatic selection. Because `\jegg*` selects the gray background rather than a sizing path, `\jegg` has no manual lowercase-sizing star when the option is off.

For lowercase roman numerals produced by `\Rrnum{n}`, note that `\Rrnum{1}`, `\Rrnum{5}` and `\Rrnum{10}` expand to a single letter and therefore take the lowercase path, while `\Rrnum{2}` and `\Rrnum{8}` do not. Do not try to even this out with explicit stars: keep `autolowercase=1`, use the non-starred commands, and add one group-wide `\kksref{\Rrnum{6}}`. See the “Roman numerals” section for the recommended usage and the verified corner cases.

Because its outline is an ellipse rather than a square, `\jegg` sizes lowercase letters by a rule of its own. The glyph is scaled uniformly, so `\jegg{c}` keeps a visibly lowercase x-height beside `\jegg{C}` instead of being stretched to cap height, and wide letters such as `\jegg{m}` and `\jegg{w}` are reduced only as far as the ellipse requires. The height is measured with the depth included, so ascenders and descenders such as `\jegg{f}` and `\jegg{g}` stay inside the outline as well. This rule applies to the egg alone; the other enclosures are unchanged, and the gray background of `\jegg*` is not affected by it.

`\jegg` の囲みは正方形ではなく楕円のため、小文字の大きさは専用の規則で決めています。縦横比を保ったまま縮小するので、`\jegg{c}` は `\jegg{C}` の隣でも小文字らしい高さを保ち、`\jegg{m}` や `\jegg{w}` のような幅の広い字も楕円に収まる範囲でのみ小さくなります。高さは深さ込みで測るため、`\jegg{f}` や `\jegg{g}` の上下のはみ出しも防がれます。この規則は `\jegg` のみに適用され、他の囲みコマンドの挙動は変わりません。`\jegg*` の網掛けにも影響しません。

**linewidth and command-specific line widths** The thickness of every TikZ-drawn enclosure can be set with `linewidth=<dimension>`. A command-specific option takes precedence over this common value:

| Option                               | Command                     |
|--------------------------------------|-----------------------------|
| <code>marulinewidth</code>           | <code>\maru</code>          |
| <code>kuromarulinewidth</code>       | <code>\kuromaru</code>      |
| <code>nmarulinewidth</code>          | <code>\nmaru</code>         |
| <code>jegglinewidth</code>           | <code>\jegg</code>          |
| <code>seihoulinewidth</code>         | <code>\seihou</code>        |
| <code>kuroseihoulinewidth</code>     | <code>\kuroseihou</code>    |
| <code>seimarulinewidth</code>        | <code>\seimaru</code>       |
| <code>kuroseimarulinewidth</code>    | <code>\kuroseimaru</code>   |
| <code>hishilinelwidth</code>         | <code>\hishi</code>         |
| <code>kurohishilinelwidth</code>     | <code>\kurohishi</code>     |
| <code>maruhishilinelwidth</code>     | <code>\maruhishi</code>     |
| <code>kuromaruhishilinelwidth</code> | <code>\kuromaruhishi</code> |

All of these options default to `auto`. The effective order is the command-specific value, then `linewidth`, then that command's historical default. Values other than `auto` must be TeX dimensions, for example `.6pt` or `.04\zw`. Explicit values are parsed when the package loads. Absolute dimensions such as `.6pt` do not follow later font-size changes; use a relative value such as `.04\zw` when proportional scaling is wanted. A malformed value produces a TeX dimension error while the package is loading. For `\nmaru`, the distance between its double rules remains `.03\zw`, so very large line widths can close that gap.

For `\jegg`, the interior its lowercase sizing aims at follows the effective line width. The stroke is centred on the same ellipse path whatever its width, so a thicker egg outline eats into the interior and lowercase letters are reduced a little further to stay clear of it. A line width wide enough to leave no interior at all draws a solid egg; the lowercase glyph then keeps its plain size rather than being scaled against a shape that no longer has an inside.

`\jegg` の小文字サイズが基準とする内側の領域は、実際に描画される線幅に追従します。線幅によらず線は楕円の経路の中心に描かれるため、線が太いほど内側は狭くなり、小文字はその分だけ小さくなります。内側が残らないほど太い線幅を指定した場合は塗り潰した楕円になり、そのときは縮小せず通常の小文字サイズを保ちます。

Input

```

1 \usepackage[
2   linewidth=.6pt,
3   marulinewidth=.3pt,
4   hishilinelwidth=1pt
5 ]{KKsymbols}
```

The bracket commands such as `\kakko` use font glyphs rather than TikZ frame paths, so these options do not affect them. The commands `\ichimoji` and `\zenkakuhabafixer` do not draw frames and are likewise unaffected.

枠線の太さは `linewidth=<寸法>` で一括指定できます。上表の個別オプションを指定したコマンドでは、その値が一括指定より優先されます。すべて `auto` の場合は従来の太さを維持します。括弧グリフを使うコマンドと、枠線を描画しないコマンドは対象外です。

**Compatibility notices** A pre-existing interface miswiring made `\maruhishi` call the filled `\kurohishi` implementation. This has been corrected, so `\maruhishi` now draws its intended unfilled rounded diamond and its output intentionally changes. Explicit stars whose public meaning is safe sizing now accept arbitrary material and use one common horizontal/vertical scale factor to keep it inside the outline actually drawn. This applies throughout the maru, seihou, hishi, bracket, and `\ichimoji` families, and follows each TikZ enclosure’s configured line width. The `\jegg` star is the exception: it keeps selecting a gray background only. A separate pre-existing state leak made one-digit `\jegg*` sizing depend on the preceding enclosure command. That leak is also fixed; because this is an unconditional correctness fix, `autolowercase=0` disables automatic lowercase selection but does not recreate the old context-dependent bug.

従来の `\maruhishi` は誤って `\kurohishi` の実装へ接続されていました。この接続を修正したため、出力は本来の白抜き角丸菱形へ変わります。また、安全サイズを意味する明示スターは任意の内容を受け取り、縦横共通の倍率で縮小して実際に描かれる枠内へ収めます。これは `maru`・`seihou`・`hishi`・括弧・`\ichimoji` の各系列に適用され、TikZ で描く枠では指定線幅にも追従します。例外として、`\jegg` のスターは従来どおり網掛けだけを選びます。また、1 桁数字の `\jegg*` の大きさが直前の囲みコマンドに依存する不具合も修正しています。この修正は `autolowercase=0` でも有効です。

## 5 Caution

Since this package internally calls `\ltjghostbeforejachar` and `\ltjghostafterjachar`, it can be used only in a LuaLaTeX environment.

## 6 Commands

### 6.1 The maru series

This package provides `\maru`, `\kuromaru`, and `\nmaru`. Each of them takes one mandatory argument and no optional arguments. You can pass strings of any length and in any font as arguments.

With the default `autolowercase=1` setting, an argument that fully expands to exactly one lowercase ASCII letter automatically uses the lowercase-safe sizing path, so `\maru{m}` is normally

sufficient. The explicit form `\maru*{m}` still forces that sizing path. With `autolowercase=0`, use the star manually as in earlier versions. For `\jegg`, the star remains reserved for changing the background to gray; lowercase sizing is detected independently only while `autolowercase=1`. With `autolowercase=0`, `\jegg` has no manual lowercase-sizing star. Protected formatting commands are not removed by this classification. If an argument such as `\textbf{m}` does not expand to the bare letter, use an explicit star on commands whose star selects the lowercase-sizing path. That explicit star is not restricted to one letter: material such as `abc` or `cmj` is uniformly shrunk in both directions until it fits the actual outline, without changing the glyphs' aspect ratio. The fit also follows the configured frame line width.

The egg keeps lowercase letters uniformly scaled inside the ellipse it draws, including wide letters and letters with ascenders or descenders:

© C   m M   w W   f F   ● ■ ● ■

Input

Automatic lowercase sizing

```

1 % Normal Characters
2 \maru{A}\maru{あ}\maru{QED.}
3
4 % Lowercase ASCII is detected automatically.
5 \maru{a}\maru{j}\maru{z}
6
7 % The star remains available as an explicit override.
8 \maru*{a}

```

Output

A あ QED.  
a j z  
a

They are used as follows.

表 2: maru series

| argument | \maru | \kuromaru | \nmaru | \jegg | \jegg* |
|----------|-------|-----------|--------|-------|--------|
| 1        | ①     | ❶         | ①      | ①     | ❶      |
| 97       | ⑨⑦    | ❹⑦        | ⑨⑦     | ⑨⑦    | ❹⑦     |
| だ        | ㊦     | ❧         | ㊦      | ㊦     | ❧      |
| ぱぱぱ      | ㊱㊲㊳   | ❨❩❪       | ㊱㊲㊳    | ㊱㊲㊳   | ❨❩❪    |
| m        | ㊴     | ❫         | ㊴      | ㊴     | ❫      |
| Qjg      | ㊵     | ❬         | ㊵      | ㊵     | ❬      |

They behave as if they were single kanji or hiragana characters:

あいう $\textcircled{\text{あ}}$ あいう $\textcircled{\text{あ}}$ ①②③あいうえお

The spacing between `\maru` and other characters is adjusted using `\ltjghostbeforejachar` and `\ltjghostafterjachar` so that it behaves like hiragana or kanji.

When changing the font size using commands such as `\Large`, each command is scaled proportionally according to the font size change:



You can also change the current font:



## 7 Roman numerals

**Japanese** `KKsymbols` 自体はローマ数字を生成しません。ローマ数字の生成には、通常 `KKran` パッケージが提供する `\Rrnum` を使います。`\Rrnum{8}` は小文字ローマ数字、`\Rrnum*{8}` は大文字ローマ数字を出します。

大文字ローマ数字は大文字アルファベットと同じ扱いで、通常は非スター版の囲みコマンドに入れます。たとえば `\kakko{\Rrnum*{8}}` や `\maru{\Rrnum*{8}}` のように書きます。

小文字ローマ数字は注意が必要です。小文字の `i`、`v`、`x` は字ごとに高さが違うため、単に `\kakko{\Rrnum{1}}` から `\kakko{\Rrnum{10}}` までを並べると、囲み内部での縦方向の見え方が完全には揃わないことがあります。これはバグではなく、引数の実際の高さが異なるためです。

そのため小文字ローマ数字については、明示スターを使わない非スター版のコマンドに、グループ内で一括指定した `\kksref{\Rrnum{6}}` を組み合わせる書き方を推奨します。`\kksref` は各囲みコマンドの内部に `\vphantom{\Rrnum{6}}` 相当の基準を加えるためのフックで、グループ内のすべての囲みに同じ基準を与えます。理由と検証結果は次の二つの小節にまとめます。

**English** `KKsymbols` itself does not generate roman numerals. In ordinary use, roman numerals are supplied by the `KKran` package through `\Rrnum`. `\Rrnum{8}` gives a lowercase roman numeral, while `\Rrnum*{8}` gives an uppercase roman numeral.

Uppercase roman numerals behave like uppercase Latin letters. They should normally be used with the non-starred enclosure commands, for example `\kakko{\Rrnum*{8}}` or `\maru{\Rrnum*{8}}`.

Lowercase roman numerals require more care. The letters `i`, `v`, and `x` have different heights, so a sequence from `\kakko{\Rrnum{1}}` to `\kakko{\Rrnum{10}}` may not look vertically identical inside the enclosures. This is not a bug; it follows from the actual glyph metrics of the arguments.

For lowercase roman numerals, therefore, the **recommended** way to write a sequence is to use the **non-starred** enclosure commands together with a **single group-wide** `\kksref{\Rrnum{6}}`.

`\kksref` is a hook that adds a reference equivalent to `\vphantom{\Rrnum{6}}` inside each enclosure command, so every enclosure in that group is built against the same reference. The reasons and the verification results are given in the two subsections below.

| Input                                                                                                                                                                                                                                                                                                                                                                                              | Roman numerals                       |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|
| <pre> 1 % Uppercase roman numerals: use the non-starred enclosure command. 2 \kakko{\Rrnum*{1}}\kakko{\Rrnum*{2}}\kakko{\Rrnum*{3}} 3 4 % Lowercase roman numerals: automatic reference only. 5 \kakko{\Rrnum{1}}\kakko{\Rrnum{5}} 6 7 % Lowercase roman numerals: force a common reference. 8 {\kksref{\Rrnum{6}}}% 9 \kakko{\Rrnum{1}}\kakko{\Rrnum{2}}\kakko{\Rrnum{3}}\kakko{\Rrnum{6}}}</pre> |                                      |
| Output                                                                                                                                                                                                                                                                                                                                                                                             |                                      |
|                                                                                                                                                                                                                                                                                                                                                                                                    | (I)(II)(III) (i)(v) (i)(ii)(iii)(vi) |

## 7.1 Recommended usage: non-starred commands and one group-wide phantom

**Japanese** 小文字ローマ数字を並べるときは、次の二点を守る書き方を推奨します。

1. 明示スターを使わず、非スター版の囲みコマンドを使う。
2. グループの先頭で `\kksref{\Rrnum{6}}` を一度だけ書き、そのグループ全体に一括で効かせる。

理由は三つあります。

第一に、既定の自動基準は現在フォントの  $x$  ハイト相当で、これは  $v$  や  $x$  自身の高さとはほぼ一致します。つまり小文字ローマ数字に対しては自動基準が実質的に働かず、 $i$  を含む字 ( $i$ ,  $ii$ ,  $iii$ ,  $iv$ ,  $vi$ ,  $vii$ ,  $viii$ ,  $ix$ ) と  $v \cdot x$  とで基準がずれたままになります。ヒラギノ明朝 Pro W3 の 10pt では、前者の基準高さが 7.61 pt、後者が 5.38 pt で、2.23 pt の差が残ります。

第二に、`\kksref{\Rrnum{6}}` をグループ内に一度書けば、そのグループ内のすべての囲みが同じ基準 ( $vi$  の高さや深さ) で組まれます。同じ条件では 10 個すべてが 7.61 pt となり、差は 0 になります。基準は現在のフォントサイズから決まるので、`\tiny` から `\Huge` までどのサイズでも差は 0 のままです。

なお、 $1$  や  $d$  を含むローマ数字 (50, 500, 1888 など) まで扱う場合は、それらが  $vi$  よりわずかに背が高いため、基準を `\kksref{\Rrnum{50}}` ( $1$ ) にしてください。

第三に、明示スターは「揃えるための指定」ではありません。スターは小文字向けの安全サイズ経路を選ぶもので、内容ごとに縮小率が決まります。全体が一段小さくなるうえ、 $i$  系と  $v \cdot x$  の見え方の差は残るため、一括 phantom の代わりにはなりません。

**English** When you typeset a sequence of lowercase roman numerals, the recommended style is:

1. use the non-starred enclosure commands, without an explicit star, and
2. write `\kksref{\Rrnum{6}}` once at the beginning of the group, so that it applies to the whole group at once.

There are three reasons.

First, the default automatic reference is the x-height of the current font, which is essentially the height of `v` and `x` themselves. For lowercase roman numerals the automatic reference therefore does almost nothing, and the numerals that contain `i` (i, ii, iii, iv, vi, vii, viii, ix) keep a different reference from `v` and `x`. With Hiragino Mincho Pro W3 at 10 pt the former are measured at 7.61 pt and the latter at 5.38 pt, a residual difference of 2.23 pt.

Second, one `\kksref{\Rrnum{6}}` inside the group makes every enclosure in that group use the same reference, namely the height and depth of `vi`. Under the same conditions all ten become 7.61 pt and the difference is 0. The reference follows the current font size, so the difference stays 0 from `\tiny` to `\Huge`.

If the range you typeset includes numerals containing `l` or `d` (50, 500, 1888, ...), use `\kksref{\Rrnum{50}}` (`l`) as the reference instead, since those letters are slightly taller than `vi`.

Third, an explicit star is not an alignment control. The star selects the lowercase-safe sizing path, whose shrink factor is decided per argument. It makes the whole sequence one step smaller while the difference between the `i` group and `v/x` remains, so it is not a substitute for the group-wide phantom.

| Input | Recommended                                                                       |
|-------|-----------------------------------------------------------------------------------|
| 1     | % Recommended: non-starred commands + one \kksref for the whole group.            |
| 2     | {\kksref{\Rrnum{6}}%                                                              |
| 3     | \maru{\Rrnum{1}}\maru{\Rrnum{2}}\maru{\Rrnum{3}}\maru{\Rrnum{4}}\maru{\Rrnum{5}}% |
| 4     | \maru{\Rrnum{6}}\maru{\Rrnum{7}}\maru{\Rrnum{8}}\maru{\Rrnum{9}}\maru{\Rrnum{10}} |
| 5     |                                                                                   |
| 6     | % Not recommended: the automatic reference alone.                                 |
| 7     | \maru{\Rrnum{1}}\maru{\Rrnum{2}}...\maru{\Rrnum{10}}                              |
| 8     |                                                                                   |
| 9     | % Not recommended: an explicit star does not align the sequence.                  |
| 10    | \maru*{\Rrnum{1}}\maru*{\Rrnum{2}}...\maru*{\Rrnum{10}}                           |

Recommended:  $\textcircled{i} \textcircled{ii} \textcircled{iii} \textcircled{iv} \textcircled{v} \textcircled{vi} \textcircled{vii} \textcircled{viii} \textcircled{ix} \textcircled{x}$

Automatic reference only:  $\textcircled{i} \textcircled{ii} \textcircled{iii} \textcircled{iv} \textcircled{v} \textcircled{vi} \textcircled{vii} \textcircled{viii} \textcircled{ix} \textcircled{x}$

Explicit star:  $\textcircled{i} \textcircled{ii} \textcircled{iii} \textcircled{iv} \textcircled{v} \textcircled{vi} \textcircled{vii} \textcircled{viii} \textcircled{ix} \textcircled{x}$

## 7.2 Corner-case test sheet

**Japanese** 以下は、推奨の書き方（非スター版のコマンド + グループ内で一括指定した  $\backslash\text{kksref}\{\text{Rrnum}\{6\}\}$ ）を、条件を変えて確かめたものです。表の数値はこのドキュメント自身がコンパイル時に実測した値で、各囲みコマンドが内部で使う基準ボックスの高さの、 $\text{Rrnum}\{1\}$  から  $\text{Rrnum}\{10\}$  まで（C2 では各集合内）の最大と最小の差です。0 なら、その集合のすべてが同じ基準で組まれていることを意味します。各記号を囲む細い枠は  $\text{fbox}$  (0.1 pt) で、記号の一部ではなく出力ボックスの実寸を示します。

**English** The following sheet checks the recommended style—non-starred commands with one group-wide  $\backslash\text{kksref}\{\text{Rrnum}\{6\}\}$ —under varying conditions. Every figure in the tables is measured by this document at compile time: it is the difference between the largest and the smallest height of the reference box that the enclosure commands build internally, taken over  $\text{Rrnum}\{1\}$  to  $\text{Rrnum}\{10\}$  (or over each set in C2). Zero means that every member of the set is built against the same reference. The thin frame around each symbol is an  $\text{fbox}$  (0.1 pt) showing the real size of the output box, not part of the symbol.

### C1. Font size / フォントサイズ

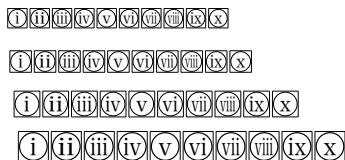
本文フォント（このドキュメントは 8pt 基準）でサイズを変えたときの差。最下行だけは 10pt を明示指定しており、これが前小節の本文で挙げた値です。

The spread at each size of the running font (this document is set at 8pt), plus one row at an explicit 10pt, which is the value quoted in the previous subsection.

| size                           | auto      | $\backslash\text{kksref}\{\text{Rrnum}\{6\}\}$ | $\backslash\text{kksrefoff}$ |
|--------------------------------|-----------|------------------------------------------------|------------------------------|
| $\text{\tiny}$                 | 0.89201pt | 0.0pt                                          | 0.89201pt                    |
| $\text{\scriptsize}$           | 1.24881pt | 0.0pt                                          | 1.24881pt                    |
| $\text{\footnotesize}$         | 1.42722pt | 0.0pt                                          | 1.42722pt                    |
| $\text{\small}$                | 1.60558pt | 0.0pt                                          | 1.60558pt                    |
| $\text{\normalsize}$           | 1.784pt   | 0.0pt                                          | 1.784pt                      |
| $\text{\large}$                | 2.1408pt  | 0.0pt                                          | 2.1408pt                     |
| $\text{\Large}$                | 2.49759pt | 0.0pt                                          | 2.49759pt                    |
| $\text{\LARGE}$                | 3.0328pt  | 0.0pt                                          | 3.0328pt                     |
| $\text{\huge}$                 | 3.568pt   | 0.0pt                                          | 3.568pt                      |
| $\text{\Huge}$                 | 4.46pt    | 0.0pt                                          | 4.46pt                       |
| $\text{\fontsize}\{10\}\{12\}$ | 2.23pt    | 0.0pt                                          | 2.23pt                       |

見たい目 ( $\backslash\text{kksref}\{\text{Rrnum}\{6\}\}$ 、 $\text{\maru}$ 、 $i\sim x$ ) :

$\textcircled{i} \textcircled{ii} \textcircled{iii} \textcircled{iv} \textcircled{v} \textcircled{vi} \textcircled{vii} \textcircled{viii} \textcircled{ix} \textcircled{x}$



## C2. Range of the argument / 引数の範囲

1 と d は vi よりわずかに背が高く、基準より高い内容は引数自身の寸法が優先されるため、それらを含む集合だけは `\kksref{\Rrnum{6}}` でも差が残ります。 `\kksref{\Rrnum{50}}` (1) を基準にすれば 0 になります。

1 and d are slightly taller than vi, and material taller than the reference keeps its own metrics, so sets containing them keep a small spread under `\kksref{\Rrnum{6}}`. Using `\kksref{\Rrnum{50}}` (1) as the reference brings them to 0 as well.

| set                 | values                                       | auto    | <code>\Rrnum{6}</code> | <code>\Rrnum{50}</code> |
|---------------------|----------------------------------------------|---------|------------------------|-------------------------|
| xi~xx               | 11~20                                        | 1.784pt | 0.0pt                  | 0.0pt                   |
| i, v, x, l, c, d, m | 1, 5, 10, 50, 100, 500, 1000                 | 1.824pt | 0.04001pt              | 0.0pt                   |
| iv~mmmdccclxxxviii  | 4, 9, 40, 49, 90, 400, 900, 1000, 1888, 3888 | 1.712pt | 0.04001pt              | 0.0pt                   |

xi~xx (11~20)

`\maru --- auto`



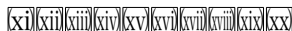
`\maru --- \kksref{\Rrnum{6}}`



`\kakko --- auto`



`\kakko --- \kksref{\Rrnum{6}}`



`\seihou --- auto`



`\seihou --- \kksref{\Rrnum{6}}`



`\hishi --- auto`



`\hishi --- \kksref{\Rrnum{6}}`



`\jegg --- auto`



XI XII XIII XIV XV XVI XVII XVIII XIX XX

xi xii xiii xiv xv xvi xvii xviii xix xx

xi xii xiii xiv XV xvi xvii xviii xix XX

aru --- auto



ivxlc dm

i

v

x

l

c

d

m

ivxldm

ivxlc dm

i v x l c d m

i v x l c d m

ivxldcm

ivxlcdm

i v x l c d m

i v x l c d m

aru --- auto

`\maru --- \kksref{\Rrnum{6}}`  

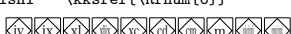

`\kakko --- auto`  


`\kakko --- \kksref{\Rrnum{6}}`  

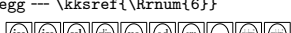

`\seihou --- auto`  

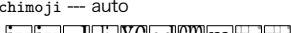

`\seihou --- \kksref{\Rrnum{6}}`  

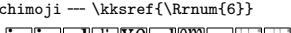

`\hishi --- auto`  


`\hishi --- \kksref{\Rrnum{6}}`  


`\jegg --- auto`  


`\jegg --- \kksref{\Rrnum{6}}`  


`\ichimoji --- auto`  


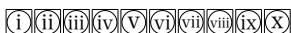
`\ichimoji --- \kksref{\Rrnum{6}}`  


### C3. With an explicit star / 明示スターとの併用

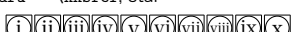
同じ内容を、スターなし／明示スターの両方で。スターでも基準は効きますが、全体が一段小さくなります。

The same material with and without an explicit star. The reference still applies under the star, but the whole sequence becomes one step smaller.

`\maru --- auto, no star`  


`\maru --- auto, star`  


`\maru --- \kksref, no star`  


`\maru --- \kksref, star`  


\kakko --- auto, no star

i ii iii iv v vi vii viii ix x

\kakko --- auto, star

i ii iii iv v vi vii viii ix x

\kakko --- \kksref, no star

i ii iii iv v vi vii viii ix x

\kakko --- \kksref, star

i ii iii iv v vi vii viii ix x

\seihou --- auto, no star

i ii iii iv v vi vii viii ix x

\seihou --- auto, star

i ii iii iv v vi vii viii ix x

\seihou --- \kksref, no star

i ii iii iv v vi vii viii ix x

\seihou --- \kksref, star

i ii iii iv v vi vii viii ix x

\hishi --- auto, no star

i ii iii iv v vi vii viii ix x

\hishi --- auto, star

i ii iii iv v vi vii viii ix x

\hishi --- \kksref, no star

i ii iii iv v vi vii viii ix x

\hishi --- \kksref, star

i ii iii iv v vi vii viii ix x

\jegg --- auto, no star

i ii iii iv v vi vii viii ix x

\jegg --- auto, star

i ii iii iv v vi vii viii ix x

\jegg --- \kksref, no star

i ii iii iv v vi vii viii ix x

\jegg --- \kksref, star

i ii iii iv v vi vii viii ix x

\ichimoji --- auto, no star

i ii iii iv v vi vii viii ix x

\ichimoji --- auto, star

i ii iii iv v vi vii viii ix x

\ichimoji --- \kksref, no star



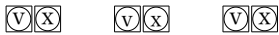
\ichimoji --- \kksref, star



## C4. Scope / スコープ

左=グループ前 (既定)、中=グループ内、右=グループ後。 \kksref の効果はグループを出ると元に戻ります。

Before the group, inside it, and after it. The effect of \kksref reverts on leaving the group.



実測 (\Rrnum{5} の基準高さ) / measured reference height of \Rrnum{5}: before 4.304pt inside 6.088pt after 4.304pt

\kksreffoff のあと \kksrefauto で復帰: default 4.304pt off 4.304pt auto 4.304pt

## C5. Typeface / 書体

upright



bold



italic



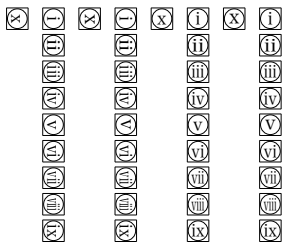
gothic



## C6. Vertical writing / 縦組み

縦組みでも同じように効きます。左から、auto、\kksref、\RotTate[0] 併用の auto、\RotTate[0] 併用の \kksref。

The hook works in vertical mode as well. From the right-hand side: auto, \kksref, then the same two with \RotTate[0].



縦組みの中で実測した基準高さ／reference heights measured inside the vertical box above: `\Rrnum{5}` default 1.272pt, with `\kksref{\Rrnum{6}}` 3.056pt; `\Rrnum{6}` default 3.056pt. 指定後の `\Rrnum{5}` が `\Rrnum{6}` の値に一致します。

## C7. Material taller than the reference / 基準より背の高い内容

漢字・かな・欧文大文字は基準より背が高く、`\kksref{\Rrnum{6}}` を掛けても出力は変わりません。数字と大文字ローマ数字については、このフォントでは `i` のドットが大文字の高さをごくわずかに上回るため、基準高さが 0.06 pt 程度上がることがあります（下の実測を参照）。紙面上は判別できない差ですが、ローマ数字用に `\kksref` を掛けたグループへ他の内容を混ぜる場合は、この点だけ意識してください。

Kanji, kana and uppercase Latin letters are taller than the reference, so `\kksref{\Rrnum{6}}` leaves their output unchanged. For digits and uppercase roman numerals the reference height can rise by about 0.06pt, because in this font the dot of `i` sits marginally above cap height; see the measurements below. The difference is not visible on the page, but it is worth knowing if you mix other material into a group whose `\kksref` was set for roman numerals.

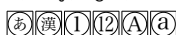
uppercase --- auto



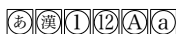
uppercase --- `\kksref`



kana, kanji, digits, latin --- auto



kana, kanji, digits, latin --- `\kksref`



実測／measured: `\Rrnum*{8}` auto 6.064pt, `kksref` 6.088pt    あ auto 7.04pt, `kksref` 7.04pt    1 auto 6.032pt, `kksref` 6.088pt    A auto 6.12pt, `kksref` 6.12pt

## C8. Package options / パッケージオプション

**Japanese** パッケージオプションは読み込み時に決まるため、この一枚では切り替えられません。次の三点は、オプションを変えた別のビルドで確認した結果です。

**tsumesuji** ローマ数字には無関係です。このオプションは 2 桁以上の数字列だけを対象とするため、`tsumesuji=0` と `tsumesuji=1` で本節の出力は完全に一致します。

**autolowercase** `autolowercase=1` (既定) では、展開結果が 1 字になる `i`, `v`, `x`, `l`, `c`, `d`, `m` が小文字経路に入り、`ii` や `viii` などは通常経路のままになります。`\kksref` はどちらの経路にも効きますが、最も揃うのは `autolowercase=1` + 非スター + 一括 `phantom` の組み合わせです。

**linewidth** 系 枠線を太くしても内容は内側に収まります。基準の効き方は線幅によって変わりません。

**English** Package options are fixed when the package is loaded, so they cannot be switched inside this sheet. The following three points were checked with separate builds.

**tsumesuji** Unrelated to roman numerals. The option applies only to digit strings of two or more digits, so the output of this section is identical under `tsumesuji=0` and `tsumesuji=1`.

**autolowercase** With `autolowercase=1` (the default), the one-letter results `i`, `v`, `x`, `l`, `c`, `d` and `m` take the lowercase path while `ii`, `viii` and the like keep the normal one. `\kksref` applies to both paths, but the most uniform combination is `autolowercase=1` with the non-starred commands and a group-wide phantom.

**linewidth keys** A heavier outline still keeps the material inside, and the reference behaves the same at any line width.

## 8 Automatic alignment (reference box)

**Japanese** v2.2.0 以降、各囲みコマンドは既定で「現在フォントの参照高さ」、具体的には `x-height` 相当を基準として中身のサイズと中央位置を決めます。これは短い文字や深さのない文字を、引数自身のバウンディングボックスだけで処理したときのばらつきを抑えるためです。

ただし、既定の自動基準は `\vphantom{\Rrnum{6}}` と同一ではありません。小文字ローマ数字については自動基準が実質的に働かないため、前節の推奨のとおり、非スター版のコマンドに `\kksref{\Rrnum{6}}` をグループ内で一括指定してください。

漢字・かな・大文字・数字など、既定基準より背の高い内容は、引数自身の寸法が優先されます。このため、通常の文書では既存の出力が大きく変わらないようになっています。

**English** Since v2.2.0, each enclosure command uses the current font's reference height, roughly its `x-height`, as the default reference for sizing and vertical centering. This reduces variation when short glyphs or glyphs with no depth would otherwise be measured only by their own bounding boxes.

However, the automatic reference is not identical to `\vphantom{\Rrnum{6}}`. For lowercase roman numerals it has essentially no effect, so follow the recommendation of the previous section: use the non-starred commands with one group-wide `\kksref{\Rrnum{6}}`.

For content taller than the default reference, such as kanji, kana, uppercase letters, or digits, the argument's own metrics still dominate. This keeps ordinary documents close to their existing appearance.

### 8.1 Overriding the reference

`\kksref{<material>}` **Japanese:** `<material>` の高さ・深さを、既定の自動基準の代わりに使います。効果は `\RotYoko` と同じくスコープ依存なので、通常はグループで囲みます。

**English:** Use the height and depth of `<material>` as the reference instead of the automatic one. The effect is scoped like `\RotYoko`, so enclose it in a group.

Input

```
1 {\kksref{\Rrnum{6}}}%  
2 \maru{\Rrnum{1}}\maru{\Rrnum{4}}\maru{\Rrnum{8}}}
```

**\kksrefoff** **Japanese:** 参照ボックスを無効化し、v2.2.0 以前のように引数自身の寸法を使います。

**English:** Disable the reference box and fall back to the argument’s own metrics, which is the behavior before v2.2.0.

**\kksrefauto** **Japanese:** 既定の自動基準に戻します。**\kksrefreset** は別名です。

**English:** Restore the automatic reference. **\kksrefreset** is an alias.

## 9 Rotation

### 9.1 Commands

This package provides **\RotTate** and **\RotYoko**. The differences are as follows:

**In horizontal mode** You should use **\RotYoko**. The default value is 0. Therefore, if you use **\RotYoko** with no arguments, this is equal to **\RotYoko[0]**

**In vertical mode** You should use **\RotTate**. The default value is 90. Therefore, if you use **\RotTate** with no arguments, this is equal to **\RotTate[90]**

From a technical perspective: Commands like **\maru** provided by this package automatically rotate their arguments based on the “current typesetting direction” Specifically, the package applies a 0-degree rotation for horizontal writing (yoko-gaki) and a 90-degree rotation for vertical writing (tate-gaki).

The commands **\RotYoko** and **\RotTate** redefine this “automatic rotation angle” to the value specified in their arguments. Consequently, the effect is persistent unless localized (similar to how font-size commands like **\small** behave). Therefore, please ensure you use appropriate scoping, such as enclosing the command within curly braces **{...}**, when applying these settings.

Input

```
1 % In horizontal mode  
2 {\RotYoko[45]\kakko{あ}\kakko{い}\kakko{う}}\par  
3 {\RotYoko[60]\kakko{1}\kakko{2}\kakko{3}}  
4  
5 % In vertical mode  
6 \parbox<t>{5\zw}{% <b> option requires lltjtext package.  
7 {\RotTate[45]\kakko{あ}\kakko{い}\kakko{う}}\par
```

```

8   {\RotTate[60]\kakko{1}\kakko{2}\kakko{3}}
9 }

```

#### Output

```

(㊦)(㊧)(㊨)
(㊩)(㊪)(㊫)

(㊬) (㊭)
(㊮) (㊯)
(㊰) (㊱)

```

## 9.2 Vertical mode

When you want to typeset ㊦, for instance, in vertical mode, you should use `\RotTate` command.

As described in the previous subsection, the effect of `\RotTate` lasts, when localized, in a certain group. So when you typeset hiragana or kanji, use it like this:

#### Input

```

1  % In vertical mode
2  \parbox<t>{5\zw}{%
3    {\RotTate[0]\kakko{あ}\kakko{い}\kakko{う}}\par
4    \kakko{1}\kakko{1}\kakko{3}
5  }

```

#### Output

```

(1) (あ)
(1) (い)
(3) (う)

```

## 10 The seihou series

The commands introduced below are used in exactly the same way as the maru series. With `autolowercase=1` (the default), single lowercase ASCII-letter arguments are detected automatically, so `\seihou{m}` uses the lowercase-safe sizing path. The explicit `\seihou*{m}` form remains available; with `autolowercase=0` it is required for the legacy manual behavior. Explicit stars may also contain multiple characters; every seihou and hishi variant then uses one uniform scale factor to

fit both width and total height/depth inside its actual outline and configured line width.

表 3: seihou series

| argument | <code>\seihou</code>                                                              | <code>\kuroseihou</code>                                                          | <code>\seimaru</code>                                                             | <code>\kuroseimaru</code>                                                         |
|----------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| l        |  |  |  |  |
| 97       |  |  |  |  |
| だ        |  |  |  |  |
| ばばば      |  |  |  |  |
| m        |  |  |  |  |
| Qjg      |  |  |  |  |

表 4: hishi series

| argument | <code>\hishi</code>                                                               | <code>\kurohishi</code>                                                           | <code>\maruhishi</code>                                                           | <code>\kuromaruhishi</code>                                                       |
|----------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| l        |  |  |  |  |
| 97       |  |  |  |  |
| だ        |  |  |  |  |
| ばばば      |  |  |  |  |
| m        |  |  |  |  |
| Qjg      |  |  |  |  |

Input

Automatic lowercase sizing

```

1 % Normal Characters
2 \seihou{A}\seihou{あ}\seihou{QED.}
3
4 % Lowercase ASCII is detected automatically.
5 \seihou{a}\seihou{j}\seihou{z}

```

Output



## 11 The kakko series

The commands introduced below are used in exactly the same way as the maru series.

With `autolowercase=1` (the default), single lowercase ASCII-letter arguments are detected automatically, so `\kakko{m}` uses the lowercase-safe sizing path. The explicit `\kakko*{m}` form remains

available; with `autolowercase=0` it is required for the legacy manual behavior. Explicit stars may contain multiple characters and keep their aspect ratio while fitting the bracket's reserved interior. The same rule applies to the other bracket commands and to `\ichimoji`. `\zenkakuhabafixer` does not take a star and is not affected by this option.

表 5: kakko series ①

| argument | \kakko | \sumikakko | \kakukakko | \kikakko | \ykakko |
|----------|--------|------------|------------|----------|---------|
| 1        | (1)    | 【1】        | [1]        | (1)      | ⟨1⟩     |
| 97       | (97)   | 【97】       | [97]       | (97)     | ⟨97⟩    |
| だ        | (だ)    | 【だ】        | [だ]        | (だ)      | ⟨だ⟩     |
| ばばば      | (ばばば)  | 【ばばば】      | [ばばば]      | (ばばば)    | ⟨ばばば⟩   |
| m        | (m)    | 【m】        | [m]        | (m)      | ⟨m⟩     |
| Qjg      | (Qjg)  | 【Qjg】      | [Qjg]      | (Qjg)    | ⟨Qjg⟩   |

表 6: kakko series ②

| argument | \nykakko | \namikakko | \kagikakko | \nkagikakko | \ichimoji | \zenkakuhabafixer |
|----------|----------|------------|------------|-------------|-----------|-------------------|
| 1        | ⟨1⟩      | {1}        | 「1」        | 『1』         | 1         | l                 |
| 97       | ⟨97⟩     | {97}       | 「97」       | 『97』        | 97        | 97                |
| だ        | ⟨だ⟩      | {だ}        | 「だ」        | 『だ』         | だ         | だ                 |
| ばばば      | ⟨ばばば⟩    | {ばばば}      | 「ばばば」      | 『ばばば』       | ばばば       | ばばば               |
| m        | ⟨m⟩      | {m}        | 「m」        | 『m』         | m         | m                 |
| Qjg      | ⟨Qjg⟩    | {Qjg}      | 「Qjg」      | 『Qjg』       | Qjg       | Qjg               |

Input

Automatic lowercase sizing

```

1 % Normal Characters
2 \kakko{A}\kakko{あ}\kakko{QED.}
3
4 % Lowercase ASCII is detected automatically.
5 \kakko{a}\kakko{j}\kakko{z}

```

Output

(A)(あ)(QED.)

(a)(j)(z)

## 11.1 Additional Description: `\ichimoji` and `\zenkakuhabafixer`

The major difference of `\ichimoji` and `\zenkakuhabafixer` is that the former changes the vertical scale to force its totalheight to `\zw`, but the latter doesn't. You can see the difference as follows:

`\ichimoji` ㊦

`\zenkakuhabafixer` ㊦

## 12 License

Released under the MIT License.

## 13 Version History

- **v1.0.0** (2025/10/03) — Initial public release.
- **v1.0.1–1.0.4** — Added `\ichimoji`; fixed various bugs.
- **v1.1.0** (2025/10/28) — Unified all commands to `zenkaku` (full-width).
- **v1.1.1** (2025/11/10) — Refined `\ichimoji` scaling logic.
- **v2.0.0** (2025/12/23) — Overhauled scaling to match OTF character quality.
- **v2.0.1** (2026/01/08) — With the update to `luatexja` version 20260107.0, the commands provided by the `KKsymbols` package now behave identically to native Japanese characters. This improvement is due to the bug fixes in `\ltjghostbeforejachar` and `\ltjghostafterjachar`. Previously, when multiple commands from this package were used consecutively, proper glue was not inserted between them; however, this issue has been resolved in this update.
- **v2.0.2** (2026/01/20) — This update fixes a critical bug where arguments of the `\kakko` command could overflow when using specific fonts (for example, Hiragino fonts with weights W4 and above).
- **v2.1.0** (2026/02/16) — In this update, the following points are changed.
  - `\period` command was deleted. It have never been used since the significance of existence had been absolutely questionable. I finally decided to delete it.
  - New package option `tsumesuji` was added.
  - New command `\zenkakuhabafixer` was added.
- **v2.1.1** (2026/02/17) — An emergency update to fix a bug where the arguments of `\kakko`, `\maru` etc. caused expansion errors.
- **v2.1.2** (2026/02/19) — The effects are applied when the argument of a command subject to the `tsumesuji` option—after expansion—is a numeric string of two or more digits and contains the digit “1”.
- **v2.1.4** (2026/04/17) — When the arguments provided by this package contain only one-digit-number, the internal resize algorithm is nullified.

- **v2.2.0 (2026/06/21)** — Automatic vertical alignment was introduced. By default, the content of each enclosure is sized and centered against the current font’s reference height (x-height) instead of relying only on the glyph’s own metrics. The new commands `\kksref`, `\kksrefoff`, and `\kksrefauto` (alias `\kksrefreset`) control this behavior. For strict lowercase roman-numeral alignment, use `\kksref{\Rrnum{6}}` explicitly.
- **v2.2.1 (2026/06/21)** — Bug fix: font-size commands inside arguments no longer cause an infinite loop in the internal one-digit/`tsumesuji` check. The visible output size is unaffected.
- **v2.2.2 (2026/06/26)** — Documentation update. The documentation now describes uppercase `\Rrnum*{n}` and lowercase `\Rrnum{n}` usage, clarifies when `\kksref{\Rrnum{6}}` is needed, and adds a Japanese/English comparison with `luatexja-otf`.
- **v2.3.0 (2026/08/20)** — Added automatic lowercase sizing (`autolowercase`), configurable frame line widths (`linewidth` and the per-command keys such as `marulinenwidth`), and the compatibility fixes described above, including the corrected `\maruhishi` wiring.

The safe-sizing route of the circle enclosures (`\maru`, `\kuromaru`, `\nmaru`) keeps the historical rule that the material’s total height stays within the square inscribed in the circle. Fitting the material’s box to the circle itself, which the rewritten routine did at first, is geometrically correct but leaves a tall narrow letter such as `j` with no clearance where its ink actually reaches: the dot and the descender sit at the horizontal centre, that is at the circle’s poles, while the box touches the outline at its corners. A second historical rule is kept as well: material wider than the inner box settles for a slightly smaller base, which is what keeps a wide letter such as `m` or `w` clear of the frame at the sides. Material of a single glyph is judged there on its widest and tallest points rather than at its box corners, since a lone glyph’s ink does not reach the corners of its own box; this is the same test the package’s ink guard applies. A string of glyphs, whose width nothing bounds, keeps the corner test. Every rule here is an upper bound, so each can only make a fit smaller than the geometry already allows, and a heavier `linewidth` tightens all of them in step with the outline. The result is that in the circle enclosures every single lowercase letter renders at exactly the size it had in v2.2.2, while multi-character material keeps the containment the rewritten routine introduced.

Specification change (documentation): the recommended way to typeset lowercase roman numerals is now the non-starred enclosure commands combined with a single group-wide `\kksref{\Rrnum{6}}`. Until this version the documentation presented `\kksref{\Rrnum{6}}` as an optional measure for strict alignment, and left the automatic reference as the ordinary route. Measurement shows that the automatic reference is the x-height of the current font and therefore has essentially no effect on lowercase roman numerals, leaving a 2.23 pt spread at 10pt, and that an explicit star only makes the whole sequence smaller without aligning it. The `\kksref{\Rrnum{6}}` route brings the spread to 0 at every font size. The “Roman numerals” section now carries a “Recommended usage” subsection and a “Corner-case test sheet” whose figures this document measures at compile time, including the note that `\kksref{\Rrnum{50}}` is needed when the range includes numerals containing 1 or d. No package code was changed

by this item.