

The xistercian package

Jonathan P. Spratte*

2021-11-23 v1.1

Contents

1	Documentation	Γ
1.1	Introduction	⋮
1.2	Macros	⋮
1.3	Options	⋮
1.4	Bold Cistercian Numerals	⋮
2	Implementation	Π
2.1	Variables	Π
2.2	Options	⋮
2.3	User macros	⋮
2.4	Parsing	⋮
2.4.1	Small Auxiliaries	⋮
2.4.2	Input Parsing	⋮
2.5	Drawing the Digits	⋮
2.6	Rebuild the Glyphs	⋮
	Index	⋈

1 Documentation

The xistercian package provides Cistercian numerals for use in \LaTeX . The name is chosen to be xistercian instead of cistercian because I've learned that a (currently) unpublished package by that name already exists and I don't want to be an evil name-stealer.

*jspratte@yahoo.de

┐	┌	└	┘	┐	┌	└	┘	┐
1	2	3	4	5	6	7	8	9
┐	┌	└	┘	┐	┌	└	┘	┐
10	20	30	40	50	60	70	80	90
┐	┌	└	┘	┐	┌	└	┘	┐
100	200	300	400	500	600	700	800	900
┐	┌	└	┘	┐	┌	└	┘	┐
1000	2000	3000	4000	5000	6000	7000	8000	9000

Figure 1: The different base glyphs in vertical mode

┐	┌	└	┘	┐	┌	└	┘	┐
1	2	3	4	5	6	7	8	9
┐	┌	└	┘	┐	┌	└	┘	┐
10	20	30	40	50	60	70	80	90
┐	┌	└	┘	┐	┌	└	┘	┐
100	200	300	400	500	600	700	800	900
┐	┌	└	┘	┐	┌	└	┘	┐
1000	2000	3000	4000	5000	6000	7000	8000	9000

Figure 2: The different base glyphs in horizontal mode

1.1 Introduction

Cistercian numerals are a system to denote the numbers from 1 to 9999 with a single glyph. They use a stem (┐) that is used as zero by this package, and add the digits to that using small tick marks, the place denoting the digits value. On the upper right means units (┐), upper left tens (┌), lower right hundreds (└), and lower left thousands (┘). Figure 1 gives an overview over the base glyphs.

In the medieval times it was quite common (according to Wikipedia more common than the vertical style) to use the Cistercian numerals horizontally instead. This package also supports that, in which case the numerals are rotated by 90 degrees counter-clockwise, see figure 2.

All glyphs have the same bounding box, are drawn using the pgf package, and cached inside TeX box registers. Therefore the performance is quite well, at least well enough to actually consider using these numerals without a major performance hit. But it still slows down the compilation, especially if the glyphs have to be redrawn often. By default they will be redrawn if the font size changed since the last usage inside the current group scope, but you can specify which font feature changes should cause a redraw.

Since Cistercian numerals quite compactly cover a big range of natural numbers they might be used to compactly keep track of running numbers. For instance, to use Cistercian numerals as page numbers you can use:

```
\pagenumbering{cistercian}
```

or to use them as footnotes:

```
\renewcommand\thefootnote{\cistercian{footnotes}}
```

The package tries to play nice on hyperref if it is loaded, but bookmarks containing material that should've been a Cistercian numeral will have a leading CISTER in front of the numeric value to give unique strings compared to `\arabic`.

Negative numbers just keep a leading `-`, and numbers with an absolute number greater than 9999 are displayed with multiple Cistercian digits. For instance `-12345678` is displayed as `-𐌹𐌶𐌿𐌺𐌰𐌽𐌰𐌶`. And in the case that four consecutive digits are zeros they get displayed as that: `𐌹𐌶𐌿𐌺` is 10000.

1.2 Macros

<code>\cistercian</code>	<code>\cistercian{⟨counter⟩}</code>
--------------------------	-------------------------------------

Prints the value of the $\text{\LaTeX 2}_{\epsilon}$ `⟨counter⟩` as a Cistercian numeral (similar to how `\roman` prints a `⟨counter⟩` as a Roman numeral). It can also be used as `\pagenumbering{cistercian}` to change the page numbering, *etc.*

<code>\cisterciannum</code>	<code>\cisterciannum{⟨integer⟩}</code>
-----------------------------	--

Prints the `⟨integer⟩` as a Cistercian numeral. The `⟨integer⟩` has to be provided as a string containing only digits and optionally a single leading `-` (spaces are ignored). This can handle integers of arbitrary size.

<code>\cisterciannumE</code>	<code>\cisterciannum{⟨integer expression⟩}</code>
------------------------------	---

Evaluates the `⟨integer expression⟩` and prints the result as a Cistercian numeral.

<code>\cisterciansetup</code>	<code>\cisterciansetup{⟨key=value,...⟩}</code>
-------------------------------	--

Can be used to locally change the options after the package was loaded. The glyphs of the Cistercian numerals aren't automatically updated when you change the setup.

<code>\cistercianredraw</code>	<code>\cistercianredraw</code>
--------------------------------	--------------------------------

<code>\cistercianredrawlazy</code>	
------------------------------------	--

When this is used the glyphs will be redrawn for the current group. The lazy variant will not immediately redraw the glyphs, but instead ensure that the next usage of a Cistercian numeral in the current scope will redraw the glyphs. The lazy variant could result in the glyphs being redrawn multiple times (if Cistercian numerals are used in different nested groups), while the normal variant might result in the glyphs being redrawn too eagerly.

<code>\cistercianstyle</code>	<code>\cistercianstyle{⟨name⟩}{⟨key=value,...⟩}</code>
-------------------------------	--

With this macro you can define a key called `⟨name⟩` that'll set the options in the `⟨key=value⟩` list if used inside `\cisterciansetup`. Only new names are allowed.

For example, the key `horizontal` is equivalently defined to

```
\cistercianstyle{horizontal}{o=h,wd=0.775em,ht=1.13ex}
```

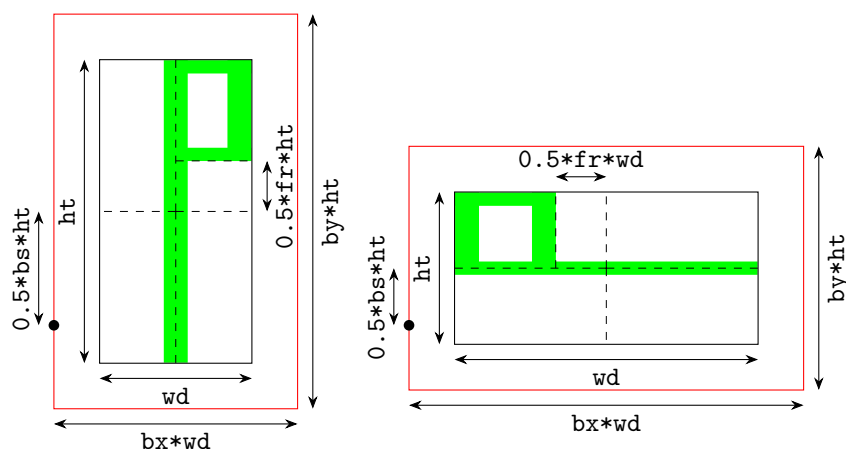


Figure 3: Measurements of a glyph. The red rectangle is the bounding box, the dot on the left shows the reference point placed on the surrounding baseline.

1.3 Options

The options described here are all usable as package options (except when explicitly stated otherwise). `xistercian` doesn't parse the global options provided to the documentclass. Additionally you can change the options (except for the debug option) using `\cisterciansetup`.

For the options `width` and `height`, and probably also the strokes, you should use dimensions using `ex` or `em` to get sizes depending on the current font size. For all three of those options the *<dimension expression>* is evaluated (using the setup specified in `font`) every time the font changed according to the `redraw` option's feature list (by default only when the size changed) since the last time they were drawn.

Initial values if present are printed on the right. A small graphic explaining most of the size related options is shown in figure 3.

<hr/> width wd <hr/>	<code>width = {<dimension expression>}</code> Set the width of the Cistercian numeral digits.	<code>.5em</code>
<hr/> height ht <hr/>	<code>height = {<dimension expression>}</code> Set the height of the Cistercian numeral digits.	<code>1.55ex</code>
<hr/> bound-x bx <hr/>	<code>bound-x = {<float>}</code> Specifies the factor the bounding box in <i>x</i> -direction is bigger than the width. An empty <i><float></i> is considered the same as 1.	<code>1.2</code>
<hr/> bound-y by <hr/>	<code>bound-y = {<float>}</code> Specifies the factor the bounding box in <i>y</i> -direction is bigger than the height. An empty <i><float></i> is considered the same as 1.	<code>empty</code>

<u>baseline</u>	baseline = $\{\langle float \rangle\}$	empty
<u>bs</u>	Sets the baseline of the symbols in multiples of the negative height. 1 (or empty) results in the symbol starting on the surrounding baseline, bigger values shift the symbols upwards, smaller values downwards.	
<u>stroke-v</u>	stroke-v = $\{\langle dimension expression \rangle\}$.18ex
<u>sv</u>	Sets the stroke thickness of vertical strokes.	
<u>stroke-h</u>	stroke-h = $\{\langle dimension expression \rangle\}$.1ex
<u>sh</u>	Sets the stroke thickness of horizontal strokes.	
<u>stroke-du</u>	stroke-du = $\{\langle dimension expression \rangle\}$.1ex
<u>sdu</u>	Sets the stroke thickness of diagonal strokes going from lower left to upper right.	
<u>stroke-dd</u>	stroke-dd = $\{\langle dimension expression \rangle\}$.18ex
<u>sdd</u>	Sets the stroke thickness of diagonal strokes going from upper left to lower right.	
<u>strokes</u>	stroke = $\{\langle dimension expression \rangle\}$	
<u>s</u>	Sets all stroke thicknesses at once.	
<u>fraction</u>	fraction = $\{\langle float \rangle\}$	/6
<u>fr</u>	fraction = $\{\langle float \rangle\}/\{\langle integer \rangle\}$	
<p>The Cistercian digits are drawn only in a fraction of the total bounding box of each symbol. The ones are drawn (with vertical Cistercians) in the upper right rectangle. The width of that rectangle is determined by width, and the upper border by height, and with this option you can set the lower border as a fraction of the height. You can either just give a factor as a $\langle float \rangle$, or just pass in a divisor as an $\langle integer \rangle$, or both. An empty $\langle float \rangle$ equals 1.</p> <p>Consider the following example (result on the right):</p> <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="flex: 1;"> <pre>{\cisterciansetup{fr=2/9,redraw}\cisterciannum{9}}\par {\cisterciansetup{fr=0.6,redraw}\cisterciannum{9}}\par</pre> </div> <div style="flex: 0.1; text-align: center;"> <p>Ɔ</p> <p>Ɔ</p> </div> </div>		
<u>orientation</u>	orientation = $\{\langle choice \rangle\}$	vertical
<u>o</u>	Sets the orientation of the Cistercian numerals. Choices are horizontal (or h for short) or vertical (or v for short). You'll most likely want to change the width and height of the symbols if you decide to change this.	
<u>horizontal</u>	horizontal	Same as setting orientation = horizontal, width = 0.775em, height = 1.13ex.
<u>vertical</u>	vertical	Same as setting orientation = vertical, width = 0.5em, height = 1.55ex.

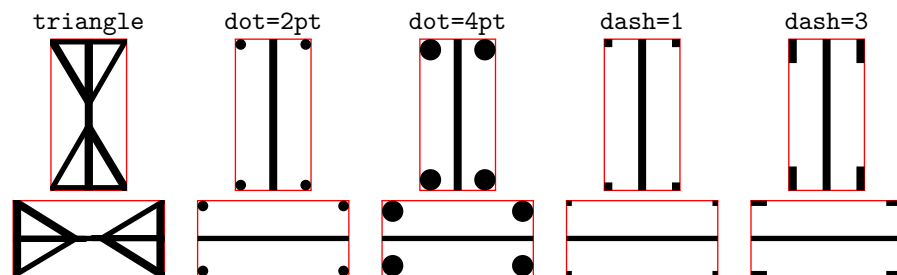


Figure 4: Alternate forms of the digit 5. The tight bounding boxes using `bx=1`, `by=1` are drawn red.

alternate-5
5

`alternate-5 = {\choice}`

`triangle`

There are alternative forms of the digit 5, while the most often used one nowadays seems to be the `triangle` form, this package also supports two other variants. Those are called `dot` and `dash`. While if you choose `triangle` you can't give an additional value, if your choice is `dot` or `dash` you can customise those using a second equals sign and some value. If you don't customise them they use their respective initial value (or last value if you changed the value at some point in time).

For `dot` you can also specify the radius as a dimension. The default is `.09ex`.

For `dash` you can specify the length of the dash in multiples of the used stroke thickness (which might differ for the horizontal and vertical symbols), an empty value is considered the same as 1. The default is *empty*.

A comparison of the three alternate forms is shown in figure 4.

The usage might look like any of the following:

```
\cisterciansetup
{
  5=triangle,
  5=dot, 5=dot=.1ex, 5={dot=.1ex},
  5=dash, 5=dash=2, 5={dash=2}
}
```

font
f

`font = {\font setup}`

`\normalfont`

Set the font which `xistercian` uses locally to evaluate the dimensions given to `width`, `height`, and `stroke` (only font switches work here, you can't use stuff like `\texttt`, instead use `\ttfamily`).

<code>redraw</code> <hr/> <code>r</code> <hr/>	<code>redraw = {\feature-list}</code> <code>redraw += {\feature-list}</code> <code>redraw</code> <code>redraw!</code>	<code>size</code>
--	--	-------------------

Selects which font features to consider when deciding to redraw the glyphs. If one of the features has changed (or the features to consider) the glyphs will be redrawn. This is especially handy if you change the font used while evaluating the given dimensions. The available features in the comma separated `\feature-list` are encoding, family, series, shape, size, and color.

If you see that a Cistercian numeral doesn't have the correct colour of the surrounding text you need to add color to the font features (xistercian uses some hack to get the boxed glyphs to respect the surrounding colour, but that hack might fail in certain circumstances).

If you use the `+=` variant of this option the `\feature-list` is added to the existing features, else they overwrite any existing configuration.

Without a value the key will issue `\cistercianredrawlazy`, and if you use the `!` variant it'll use `\cistercianredraw`. This usage is only available after the package was loaded.

<code>debug</code> <hr/>	<code>debug</code>
-----------------------------	--------------------

Only usable as a package option. If this is used a bit of debugging information is printed in the log file and terminal, and some keys try to evaluate the given argument on the spot instead of lazily. This might help to locate faulty input.

1.4 Bold Cistercian Numerals

While the package doesn't directly support bold glyphs, it is possible to utilize the hooks provided by the NFSS to pick different stroke widths whenever the font is changed to `bfseries` and back to `mdseries`:

```
\AddToHook{bfseries}
{%
  \cisterciansetup{sv=.27ex,sdd=.27ex,sh=.15ex,sdu=.15ex,wd=.625em}%
  \cistercianredrawlazy
}
\AddToHook{mdseries}
{%
  \cisterciansetup{sv=.18ex,sdd=.18ex,sh=.1ex,sdu=.1ex,wd=.5em}%
  \cistercianredrawlazy
}
```

This results in `\cisterciannum{2}` different `\enquote{series}.\par`
`\bfseries`

This results in `\cisterciannum{2}` different `\enquote{series}`.

This results in † different “series”.

This results in † different “series”.

Of course, instead of doing `\cistercianredrawlazy` in the above code, we could as well do the following equivalent thing and let `xistercian` determine when to redraw:

```
\AddToHook{bfseries}
  {\cisterciansetup{sv=.27ex,sdd=.27ex,sh=.15ex,sdu=.15ex,wd=.625em}}
\AddToHook{mdseries}
  {\cisterciansetup{sv=.18ex,sdd=.18ex,sh=.1ex,sdu=.1ex,wd=.5em}}
\cisterciansetup{redraw+=series}
```


2 Implementation

First we load the required packages

```
1 \RequirePackage{pgf,expkv-opt}
```

2.1 Variables

There are variables stored as macros to be evaluated later, as well as register types for faster access during the glyph drawing.

```
2 \newcommand*\xister@th{.1ex}
3 \newcommand*\xister@tv{.18ex}
4 \newcommand*\xister@tdu{.1ex}
5 \newcommand*\xister@td{.18ex}
6 \newcommand*\xister@x{.5em}
7 \newcommand*\xister@y{1.55ex}
8 \newcommand*\xister@bx{1.2}
9 \newcommand*\xister@by{}
10 \newcommand*\xister@bs{}
11 \newcommand*\xister@font{\normalfont}
12 \newcommand*\xister@five@dot{.09ex}
13 \newcommand*\xister@five@dash{}
14 \newcommand*\xister@share@div{/6}
15 \newcommand*\xister@share@mul{}
16 \newif\ifxister@dbg@
17 \newif\ifxister@five@triangle@\xister@five@triangle@true
18 \newif\ifxister@five@dot@
19 \newdimen\xister@X
20 \newdimen\xister@Y
21 \newdimen\xister@XY@share
22 \newdimen\xister@Th
23 \newdimen\xister@Tv
24 \newdimen\xister@Tdu
25 \newdimen\xister@Tdd
26 \newdimen\xister@TMPA
27 \newdimen\xister@TMPB
```

We also need a bunch of box registers (37 of them to be precise). For the loop we borrow an `expl3` function.

```
28 \ExplSyntaxOn
29 \int_step_inline:nn {9}
30 {
31   \expandafter\newsavebox\csname xister@digitbox@#1\endcsname
32   \expandafter\newsavebox\csname xister@digitbox@#10\endcsname
33   \expandafter\newsavebox\csname xister@digitbox@#100\endcsname
34   \expandafter\newsavebox\csname xister@digitbox@#1000\endcsname
35 }
36 \ExplSyntaxOff
37 \expandafter\newsavebox\csname xister@digitbox@0\endcsname
```

```
\xister@dbg@dim
\xister@dbg@float
```

These are initially just `\@gobble`, but the debug option might change things.

```
38 \let\xister@dbg@dim\@gobble
39 \let\xister@dbg@div\@gobble
40 \let\xister@dbg@float\@gobble
```

(End definition for `\xister@dbg@dim` and `\xister@dbg@float`.)

2.2 Options

Some macros are required to parse options.

`\xister@if@slash` We need to test whether a slash is inside an argument to parse some user option. Since this isn't needed inside an inner loop we don't create an optimised version and only borrow a generic `expl3` function.

```

41 \ExplSyntaxOn
42 \cs_new_protected:Npn \xister@if@slash #1 { \tl_if_in:nnTF {#1} { / } }
43 \ExplSyntaxOff

```

(End definition for `\xister@if@slash`.)

`\xister@fraction`

```

44 \protected\long\def\xister@fraction#1/#2\xister@stop
45 {%
46   \edef\xister@share@mul{\unexpanded{#1}}%
47   \xister@dbg@float{#1}%
48   \edef\xister@share@div{/unexpanded{#2}}%
49   \xister@dbg@div{#2}%
50 }

```

(End definition for `\xister@fraction`.)

`\xister@build` First we set the length registers to the correct size, then we draw the glyphs batch wise.
`\xister@build@h` We have to flip the x -direction for the tenths and thousands, and the y -direction for
`\xister@build@horizontal` the hundreds and thousands. The macro `\xister@sgn` is used to set the baseline in
`\xister@build@v` the correct direction. We locally disable `\pgfsetcolor` to allow building colour-unsafe
`\xister@build@vertical` boxes. This only works as long as `luacolor` (or something equivalent) isn't used, so
consider this a crude hack.

```

51 \protected\def\xister@build@vertical
52 {%
53   \let\xister@pgfsetcolor\pgfsetcolor
54   \let\pgfsetcolor@gobble
55   \xister@setlengths
56   \def\xister@sgn{-}%
57   \xister@drawzero@vertical
58   \xister@drawdigits@vertical{}{}{}%
59   \xister@X=-\xister@X
60   \xister@Y=-\xister@Y
61   \def\xister@sgn{+}%
62   \xister@drawdigits@vertical{000}{-}{-}%
63   \xister@TMPA=\xister@Tdd
64   \xister@Tdd=\xister@Tdu
65   \xister@Tdu=\xister@TMPA
66   \xister@X=-\xister@X
67   \xister@drawdigits@vertical{00}{-}{-}%
68   \xister@X=-\xister@X
69   \xister@Y=-\xister@Y
70   \def\xister@sgn{-}%
71   \xister@drawdigits@vertical{0}{-}{-}%
72   \let\pgfsetcolor\xister@pgfsetcolor
73 }
74 \protected\def\xister@build@horizontal

```

```

75  {%
76  \let\xister@pgfsetcolor\pgfsetcolor
77  \let\pgfsetcolor\@gobble
78  \xister@setlengths
79  \def\xister@sgn{-}%
80  \xister@drawzero@horizontal
81  \xister@drawdigits@horizontal{}{}{}%
82  \xister@X=-\xister@X
83  \xister@Y=-\xister@Y
84  \def\xister@sgn{+}%
85  \xister@drawdigits@horizontal{000}{-}{-}%
86  \xister@TMPA=\xister@Tdd
87  \xister@Tdd=\xister@Tdu
88  \xister@Tdu=\xister@TMPA
89  \xister@X=-\xister@X
90  \xister@drawdigits@horizontal{0}{-}{}%
91  \xister@X=-\xister@X
92  \xister@Y=-\xister@Y
93  \def\xister@sgn{-}%
94  \xister@drawdigits@horizontal{00}{-}{-}%
95  \let\pgfsetcolor\xister@pgfsetcolor
96  }
97  \let\xister@build\xister@build@vertical
98  \let\xister@build@v\xister@build@vertical
99  \let\xister@build@h\xister@build@horizontal

```

(End definition for \xister@build and others.)

\xister@fontfeatures
\xister@add@fontfeature

This macro should store the different font features which should be kept track of. It will be redefined programmatically but the default is quite easy, just track the font size.

```

100 \protected\long\def\xister@add@fontfeature#1%
101  {%
102  \begingroup\expandafter\expandafter\expandafter\endgroup
103  \expandafter\ifx
104  \csname xister@add@fontfeature@\detokenize{#1}\endcsname
105  \relax
106  \PackageError{xistercian}{Unsupported font feature ‘\detokenize{#1}’.}{}%
107  \else
108  \csname xister@add@fontfeature@\detokenize{#1}\endcsname
109  \fi
110  }
111  \def\xister@fontfeatures{\unexpanded\expandafter{\f@size}/}

```

We know that the colour hack breaks if luacolor is loaded, so if that is found to be the case we add the color feature to the list of things we need to track. But we only want to add it if it's not yet in the list, so we run a check we borrow from expl3.

```

112 \ExplSyntaxOn
113 \AddToHook{begindocument/before}
114 {
115  \@ifpackageloaded { luacolor }
116  {
117  \exp_args:No \str_if_in:nnF \xister@fontfeatures { color }
118  { \xister@add@fontfeature{color} }
119  }

```

```

120     {}
121   }
122   \ExplSyntaxOff

```

(End definition for \xister@fontfeatures and \xister@add@fontfeature.)

```

\xister@add@fontfeature@encoding
\xister@add@fontfeature@family
\xister@add@fontfeature@series
\xister@add@fontfeature@shape
\xister@add@fontfeature@size

```

These macros all just add a specific macro to the list of things contained in \xister@fontfeatures, which determines what to consider deciding to redraw.

```

123 \def\xister@tmp#1#2%
124   {%
125     \protected\expandafter\def\csname xister@add@fontfeature@#2\endcsname
126     {%
127       \edef\xister@fontfeatures
128         {%
129           \unexpanded\expandafter
130             {\xister@fontfeatures\unexpanded\expandafter{#1}}/%
131         }%
132     }%
133   }
134   \expandafter\xister@tmp\csname \@backslashchar color@.\endcsname{color}
135   \xister@tmp\f@encoding{encoding}
136   \xister@tmp\f@family {family}
137   \xister@tmp\f@series {series}
138   \xister@tmp\f@shape {shape}
139   \xister@tmp\f@size {size}

```

(End definition for \xister@add@fontfeature@encoding and others.)

Now we define the keys

```

140 \ekvifdefinedset{xister}
141   {\PackageError{xistercian}{keyval conflict detected. Aborting}{ }\endinput}{ }
142 \def\xister@tmp#1#2#3%
143   {%
144     \protected\long\ekvdef{xister}{#1}{#3}%
145     \ekvletkv{xister}{#2}{xister}{#1}%
146   }
147   \xister@tmp{wd}{width}{\edef\xister@x{\unexpanded{#1}}\xister@dbg@dim{#1}}
148   \xister@tmp{ht}{height}{\edef\xister@y{\unexpanded{#1}}\xister@dbg@dim{#1}}
149   \xister@tmp{sv}{stroke-v}{\edef\xister@tv{\unexpanded{#1}}\xister@dbg@dim{#1}}
150   \xister@tmp{sh}{stroke-h}{\edef\xister@th{\unexpanded{#1}}\xister@dbg@dim{#1}}
151   \xister@tmp{sdu}{stroke-du}
152     {\edef\xister@tdu{\unexpanded{#1}}\xister@dbg@dim{#1}}
153   \xister@tmp{sdd}{stroke-dd}
154     {\edef\xister@tdd{\unexpanded{#1}}\xister@dbg@dim{#1}}
155   \xister@tmp{bx}{bound-x}{\edef\xister@bx{\unexpanded{#1}}\xister@dbg@float{#1}}
156   \xister@tmp{by}{bound-y}{\edef\xister@by{\unexpanded{#1}}\xister@dbg@float{#1}}
157   \xister@tmp{bs}{baseline}{\edef\xister@bs{\unexpanded{#1}}\xister@dbg@float{#1}}
158   \xister@tmp{s}{strokes}
159     {%
160       \edef\xister@tv{\unexpanded{#1}}%
161       \let\xister@th\xister@tv
162       \let\xister@tdu\xister@tv
163       \let\xister@tdd\xister@tv
164       \xister@dbg@dim{#1}%
165     }

```

```

166 \xister@tmp{f}{font}{\edef\xister@font{\unexpanded{#1}}}
167 \xister@tmp{o}{orientation}
168 {%
169   \begingroup\expandafter\expandafter\expandafter\endgroup
170   \expandafter\ifx\csname xister@build@#1\endcsname\relax
171     \PackageError{xister}%
172     {Unaccepted orientation. Choices are horizontal/h or vertical/v}{}%
173   \else
174     \expandafter\let\expandafter\xister@build\csname xister@build@#1\endcsname
175   \fi
176 }
177 \xister@tmp{fr}{fraction}
178 {%
179   \xister@if@slash{#1}%
180   {\xister@fraction#1\xister@stop}%
181   {%
182     \let\xister@share@div\@empty
183     \edef\xister@share@mul{\unexpanded{#1}}%
184     \xister@dbg@float{#1}%
185   }%
186 }
187 \xister@tmp{r}{redraw}
188 {\let\xister@fontfeatures\@empty\ekvcsvloop\xister@add@fontfeature{#1}}
189 \xister@tmp{r+}{redraw+}{\ekvcsvloop\xister@add@fontfeature{#1}}
190 \ekvletkv{xister}{r+}{xister}{r+}
191 \ekvletkv{xister}{redraw+}{xister}{redraw+}
192 \protected\long\ekvdef{xister/5}{dot}
193 {%
194   \xister@five@triangle@false
195   \xister@five@dot@true
196   \edef\xister@five@dot{\unexpanded{#1}}%
197   \xister@dbg@dim{#1}%
198 }
199 \protected\long\ekvdef{xister/5}{dash}
200 {%
201   \xister@five@triangle@false
202   \xister@five@dot@false
203   \edef\xister@five@dash{\unexpanded{#1}}%
204   \xister@dbg@float{#1}%
205 }
206 \protected\ekvdefNoVal{xister/5}{triangle}{\xister@five@triangle@true}
207 \protected\ekvdefNoVal{xister/5}{dot}
208 {%
209   \xister@five@triangle@false
210   \xister@five@dot@true
211 }
212 \protected\ekvdefNoVal{xister/5}{dash}
213 {%
214   \xister@five@triangle@false
215   \xister@five@dot@false
216 }
217 \protected\ekvsetdef\xister@five@set{xister/5}
218 \ekvlet{xister}{5}\xister@five@set
219 \ekvlet{xister}{alternate-5}\xister@five@set

```

```

220 \let\xister@five@set\xister@undefined
221 \protected\ekvdefNoVal\xister}{horizontal}
222 {%
223   \let\xister@build\xister@build@horizontal
224   \def\xister@x{.775em}%
225   \def\xister@y{1.13ex}%
226 }
227 \protected\ekvdefNoVal\xister}{vertical}
228 {%
229   \let\xister@build\xister@build@vertical
230   \def\xister@x{.5em}%
231   \def\xister@y{1.55ex}%
232 }

```

The debug option redefines the debug macros to the correct definition.

```

233 \protected\ekvdefNoVal\xister}{debug}
234 {%
235   \xister@dbg@true
236   \protected\long\def\xister@dbg@dim##1%
237   {%
238     \typeout
239     {%
240       Package xistercian Debug: Trying to use dimension
241       ‘\unexpanded{##1}’.%
242     }%
243     \xister@TMPA=\dimexpr##1\relax
244     \typeout{Package xistercian Debug: Done.}%
245   }%
246   \protected\long\def\xister@dbg@float##1%
247   {%
248     \typeout
249     {Package xistercian Debug: Trying to use float ‘\unexpanded{##1}’.%}
250     \xister@TMPA=##1\z@
251     \typeout{Package xistercian Debug: Done.}%
252   }%
253   \protected\long\def\xister@dbg@div##1%
254   {%
255     \typeout
256     {Package xistercian Debug: Trying to use divisor ‘\unexpanded{##1}’.%}
257     \xister@TMPA=\dimexpr\z@/##1\relax
258     \typeout{Package xistercian Debug: Done.}%
259   }%
260 }

```

and parse the package options (and since the used names width, height, and debug are quite frequent we don’t look at the global options).

```

261 \ekvoProcessLocalOptions\xister}

```

After the package was loaded we add the two redraw variants without a value.

```

262 \protected\ekvdefNoVal\xister}{r!}{\cistercianredraw}
263 \protected\ekvdefNoVal\xister}{r} {\cistercianredrawlazy}
264 \ekvletkvNoVal\xister}{redraw} {\xister}{r}
265 \ekvletkvNoVal\xister}{redraw!}{\xister}{r!}

```

The debug option is only available as a package option, we let it throw an error:

```

266 \ekvdefNoVal{xister}{debug}
267   {\PackageError{xister}{'debug' is only available as a load time option}{}}

```

2.3 User macros

\cisterciansetup Just as a simple way to pick options later on.

```

268 \protected\ekvsetdef\cisterciansetup{xister}

```

(End definition for \cisterciansetup. This function is documented on page 1.)

\cistercianstyle We allow users to define a style they can call with a single key name inside \cisterciansetup. To achieve that we just define a NoVal key that'll call a nested \cisterciansetup.

```

269 \protected\long\def\cistercianstyle#1#2%
270   {%
271     \ekvifdefinedNoVal{xister}{#1}%
272     {\PackageError{xister}{Key '#1' already defined}{}}%
273     {\protected\ekvdefNoVal{xister}{#1}{\cisterciansetup{#2}}}%
274   }

```

(End definition for \cistercianstyle. This function is documented on page 1.)

\cistercian The macro is equivalent to other L^AT_EX counter formatting macros, so we let it build the counter name and forward that to the canonically named auxiliary macro \cistercian.

\@cistercian

```

275 \newcommand\cistercian[1]{\expandafter\@cistercian\csname c@#1\endcsname}

```

The internal just gets the current value of a counter and forwards it.

```

276 \newcommand\@cistercian[1]{\expandafter\cisterciannum\expandafter{\the#1}}

```

(End definition for \cistercian and \@cistercian. These functions are documented on page 1.)

\cisterciannum

\cisterciannum@pdf

```

277 \edef\cisterciannum
278   {%
279     \unexpanded{\xister@texorpdf}%
280     \unexpanded\expandafter
281     {\csname cisterciannum \endcsname\cisterciannum@pdf}%
282   }
283 \protected\expandafter\def\csname cisterciannum \endcsname#1{\xister@a#1@}
284 \def\cisterciannum@pdf#1{CISTER#1}

```

(End definition for \cisterciannum and \cisterciannum@pdf. These functions are documented on page 1.)

\cisterciannumE

```

285 \newcommand\cisterciannumE[1]
286   {\expandafter\cisterciannum\expandafter{\the\numexpr#1\relax}}

```

(End definition for \cisterciannumE. This function is documented on page 1.)

\cistercianredraw

\cistercianredrawlazy

```

287 \protected\def\cistercianredraw
288   {%
289     \def\xister@last@font{\xister@last@font}%
290     \xister@ensure@current
291   }
292 \protected\def\cistercianredrawlazy{\def\xister@last@font{\xister@last@font}}

```

(End definition for `\cistercianredraw` and `\cistercianredrawlazy`. These functions are documented on page 1.)

2.4 Parsing

2.4.1 Small Auxiliaries

`\xister@gobbletoat` These are just small functions gobbling some tokens.
`\xister@gobbletozero` 293 `\def\xister@gobbletoat#1@{}`
`\xister@gobblecopy` 294 `\def\xister@gobbletozero#10{}`
`\xister@done` 295 `\def\xister@gobblecopy0\rlap#1{}`
296 `\def\xister@done#1\xister@symbols{}`

(End definition for `\xister@gobbletoat` and others.)

`\xister@texorpdf`
297 `\let\xister@texorpdf \@firstoftwo`
298 `\AddToHook{package/hyperref/after}`
299 `{%`
300 `\ifdefined\texorpdfstring`
301 `\def\xister@texorpdf{\texorpdfstring}%`
302 `\fi`
303 `}`
304 `\AddToHook{begindocument/before}`
305 `{%`
306 `\ifdefined\texorpdfstring`
307 `\def\xister@texorpdf{\texorpdfstring}%`
308 `\fi`
309 `}`

(End definition for `\xister@texorpdf`.)

2.4.2 Input Parsing

`\xister@a` First we have to make sure that we're in horizontal mode, else the `\rlaps` used to insert
`\xister@b` the glyphs will create havoc. We also need to ensure that the boxes contain the correct
`\xister@c` glyphs according to the current size and colour. After that we check whether we have
`\xister@d` to step over a minus sign and go to the next step.

310 `\def\xister@a#1%`
311 `{%`
312 `\leavevmode`
313 `\xister@ensure@current`
314 `\ifx-#1-\expandafter\xister@b`
315 `\else\expandafter\xister@b\expandafter#1\fi`
316 `}`

Next we check whether the remaining number is empty. If that is the case we use `\the\numexpr\relax` to throw a missing number error and produce just a zero stem. Else we do a dry run over the input number to see how many leading zeros we must pad to get blocks of four. The two blocks of `@@@` serve as end markers since each step of both `\xister@c` and `\xister@symbols` will grab four digits.

317 `\def\xister@b#1@%`
318 `{%`
319 `\if\relax\detokenize{#1}\relax`


```

320 \expandafter\xister@gobbletozero\the\numexpr\relax
321 \xister@symbols0000@@@%
322 \expandafter\@gobbletwo
323 \fi
324 \@firstofone{\xister@c#1@@@@relax#1@@@}%
325 }
326 \def\xister@c#1#2#3#4%
327 {%
328 \xister@gobbletoat
329 #1\xister@d{}%
330 #2\xister@d{000}%
331 #3\xister@d{00}%
332 #4\xister@d0%
333 @\xister@c
334 }
335 \def\xister@d#1#2@\xister@c#3\relax{\xister@symbols#1}

```

(End definition for \xister@a and others.)

\xister@symbols When this loop is done all arguments will be @, but we only check the first here. We generate the symbols by overlaying up to five boxes, one for each decade plus the zero stem. Each digit might be skipped if it is 0. Next grab the next four digits.

```

336 \def\xister@symbols#1%
337 {%
338 \protected\def\xister@symbols##1##2##3##4%
339 {%
340 \xister@gobbletoat##1\xister@done @%
341 \xister@gobbletozero##1\xister@gobblecopy0%
342 \rlap{\copy\csname xister@digitbox@##1000\endcsname}%
343 \xister@gobbletozero##2\xister@gobblecopy0%
344 \rlap{\copy\csname xister@digitbox@##200\endcsname}%
345 \xister@gobbletozero##3\xister@gobblecopy0%
346 \rlap{\copy\csname xister@digitbox@##30\endcsname}%
347 \xister@gobbletozero##4\xister@gobblecopy0%
348 \rlap{\copy\csname xister@digitbox@##4\endcsname}%
349 \copy#1%
350 \xister@symbols
351 }%
352 }
353 \expandafter\xister@symbols\csname xister@digitbox@0\endcsname

```

(End definition for \xister@symbols.)

2.5 Drawing the Digits

\xister@clip@v Some of the digit glyphs need a bit of clipping to look nice. Let's define a small auxiliary to select a clipping region.

```

354 \newcommand*\xister@clip@v[1]
355 {%
356 \pgfpathrectanglecorners
357 {\pgfpoint{\dimexpr\ifdim\xister@X<\z@-\fi\xister@Tv\relax}{#1}}%
358 {\pgfpoint\xister@X\xister@Y}%
359 \pgfusepath{clip}%
360 }

```

```

361 \newcommand*\xister@clip@h[1]
362 {%
363   \pgfpathrectanglecorners
364     {\pgfpoint{#1}{-\dimexpr\ifdim\xister@Y<\z@-\fi\xister@Th\relax}}%
365     {\pgfpoint{-\xister@X}\xister@Y}%
366   \pgfusepath{clip}%
367 }

```

(End definition for \xister@clip@v and \xister@clip@h.)

\xister@drawdigits@vertical
\xister@drawdigits@horizontal

The digit glyphs. Since each numeral is build from the zero stem and some appendix to it this only specifies the form of the appendices. The zero stem is defined a bit down the road. #1 will determine the decade, and #2 will be either - or empty.

```

368 \newcommand\xister@drawdigits@vertical[3]
369 {%
370   \xister@XY@share=\dimexpr\xister@share@mul\xister@Y\xister@share@div\relax
371   \xister@drawdigit{1#1}%
372   {%
373     \pgfsetlinewidth{2\xister@Th}%
374     \xister@TMPA=\dimexpr\xister@Y\xister@sgn\xister@Th\relax
375     \pgfpathmoveto{\pgfpoint\z@\xister@TMPA}%
376     \pgfpathlineto{\pgfpoint\xister@X\xister@TMPA}%
377   }%
378   \xister@drawdigit{2#1}%
379   {%
380     \pgfsetlinewidth{2\xister@Th}%
381     \xister@TMPA=\dimexpr\xister@XY@share+#2\xister@Th\relax
382     \pgfpathmoveto{\pgfpoint\z@\xister@TMPA}%
383     \pgfpathlineto{\pgfpoint\xister@X\xister@TMPA}%
384   }%
385   \xister@drawdigit{3#1}%
386   {%
387     \xister@TMPB=#3\dimexpr\xister@Tv-\xister@Tdd\relax
388     \xister@clip@v\xister@XY@share
389     \pgfsetlinewidth{2\xister@Tdd}%
390     \pgfpathmoveto{\pgfpoint\xister@TMPB\xister@XY@share}%
391     \pgfpathlineto{\pgfpoint\xister@TMPB\xister@Y}%
392     \pgfpathlineto{\pgfpoint\xister@X\xister@XY@share}%
393   }%
394   \xister@drawdigit{4#1}%
395   {%
396     \xister@clip@v\z@
397     \xister@TMPB=#3\dimexpr\xister@Tv-\xister@Tdu\relax
398     \pgfsetlinewidth{2\xister@Tdu}%
399     \pgfpathmoveto{\pgfpoint\xister@TMPB\xister@Y}%
400     \pgfpathlineto{\pgfpoint\xister@TMPB\xister@XY@share}%
401     \pgfpathlineto{\pgfpoint\xister@X\xister@Y}%
402   }%
403   \xister@drawdigit{5#1}%
404   {%
405     \ifxister@five@triangle@
406       \xister@clip@v\z@
407       \xister@TMPB=%
408       #3\dimexpr\xister@Tv-\xister@Tdu\relax

```

```

409         \pgfsetlinewidth{2\xister@Tdu}%
410         \pgfpathmoveto{\pgfpoint\xister@TMPB\xister@Y}%
411         \pgfpathlineto{\pgfpoint\xister@TMPB\xister@XY@share}%
412         \pgfpathlineto{\pgfpoint\xister@X\xister@Y}%
413         \pgfusepath{stroke,clip}%
414         \pgfsetlinewidth{2\xister@Th}%
415         \xister@TMPA=\dimexpr\xister@Y\xister@sgn\xister@Th\relax
416         \pgfpathmoveto{\pgfpoint\z@\xister@TMPA}%
417         \pgfpathlineto{\pgfpoint\xister@X\xister@TMPA}%
418     \else\ifxister@five@dot@
419         \pgfpathcircle
420             {\pgfpoint{\dimexpr\xister@X-#3\xister@five@dot\relax}%
421              {\dimexpr\xister@Y-#2\xister@five@dot\relax}}%
422             {\xister@five@dot}%
423         \pgfusepath{fill}%
424     \else
425         \xister@TMPB=\dimexpr\xister@X-#3\xister@Tv\relax
426         \pgfsetlinewidth{2\xister@Tv}%
427         \pgfpathmoveto{\pgfpoint\xister@TMPB\xister@Y}%
428         \pgfpathlineto
429             {%
430                 \pgfpoint
431                 \xister@TMPB
432                 {\dimexpr\xister@Y-#2\xister@five@dash\xister@Tv*2\relax}%
433             }%
434         \fi\fi
435     }%
436 \xister@drawdigit{6#1}%
437 {%
438     \pgfsetlinewidth{2\xister@Tv}%
439     \xister@TMPB=\dimexpr\xister@X-#3\xister@Tv\relax
440     \pgfpathmoveto{\pgfpoint\xister@TMPB\xister@Y}%
441     \pgfpathlineto{\pgfpoint\xister@TMPB\xister@XY@share}%
442 }%
443 \xister@superimpose{7#1}%
444 {%
445     \rlap{\copy\csname xister@digitbox@1#1\endcsname}%
446     \copy\csname xister@digitbox@6#1\endcsname
447 }%
448 \xister@superimpose{8#1}%
449 {%
450     \rlap{\copy\csname xister@digitbox@2#1\endcsname}%
451     \copy\csname xister@digitbox@6#1\endcsname
452 }%
453 \xister@superimpose{9#1}%
454 {%
455     \rlap{\copy\csname xister@digitbox@1#1\endcsname}%
456     \rlap{\copy\csname xister@digitbox@2#1\endcsname}%
457     \copy\csname xister@digitbox@6#1\endcsname
458 }%
459 }
460 \newcommand\xister@drawdigits@horizontal[3]
461 {%
462     \xister@XY@share=-\dimexpr\xister@share@mul\xister@X\xister@share@div\relax

```

```

463 \xister@drawdigit{1#1}%
464 {%
465 \pgfsetlinewidth{2\xister@Tv}%
466 \xister@TMPA=-\dimexpr\xister@X-#3\xister@Tv\relax
467 \pgfpathmoveto{\pgfpoint\xister@TMPA\z@}%
468 \pgfpathlineto{\pgfpoint\xister@TMPA\xister@Y}%
469 }%
470 \xister@drawdigit{2#1}%
471 {%
472 \pgfsetlinewidth{2\xister@Tv}%
473 \xister@TMPA=-\dimexpr#3\xister@Tv-\xister@XY@share\relax
474 \pgfpathmoveto{\pgfpoint\xister@TMPA\z@}%
475 \pgfpathlineto{\pgfpoint\xister@TMPA\xister@Y}%
476 }%
477 \xister@drawdigit{3#1}%
478 {%
479 \xister@TMPB=#2\dimexpr\xister@Th-\xister@Tdu\relax
480 \xister@clip@h\xister@XY@share
481 \pgfsetlinewidth{2\xister@Tdu}%
482 \pgfpathmoveto{\pgfpoint\xister@XY@share\xister@TMPB}%
483 \pgfpathlineto{\pgfpoint{-\xister@X}\xister@TMPB}%
484 \pgfpathlineto{\pgfpoint\xister@XY@share\xister@Y}%
485 }%
486 \xister@drawdigit{4#1}%
487 {%
488 \xister@clip@h\z@
489 \xister@TMPB=#2\dimexpr\xister@Th-\xister@Tdd\relax
490 \pgfsetlinewidth{2\xister@Tdd}%
491 \pgfpathmoveto{\pgfpoint\xister@X\xister@TMPB}%
492 \pgfpathlineto{\pgfpoint\xister@XY@share\xister@TMPB}%
493 \pgfpathlineto{\pgfpoint{-\xister@X}\xister@Y}%
494 }%
495 \xister@drawdigit{5#1}%
496 {%
497 \ifxister@five@triangle@
498 \xister@clip@h\z@
499 \xister@TMPB=#2\dimexpr\xister@Th-\xister@Tdd\relax
500 \pgfsetlinewidth{2\xister@Tdd}%
501 \pgfpathmoveto{\pgfpoint{-\xister@X}\xister@TMPB}%
502 \pgfpathlineto{\pgfpoint\xister@XY@share\xister@TMPB}%
503 \pgfpathlineto{\pgfpoint{-\xister@X}\xister@Y}%
504 \pgfusepath{stroke,clip}%
505 \pgfsetlinewidth{2\xister@Tv}%
506 \xister@TMPA=-\dimexpr\xister@X-#3\xister@Tv\relax
507 \pgfpathmoveto{\pgfpoint\xister@TMPA\z@}%
508 \pgfpathlineto{\pgfpoint\xister@TMPA\xister@Y}%
509 \else\ifxister@five@dot@
510 \pgfpathcircle
511 {\pgfpoint{-\dimexpr\xister@X-#3\xister@five@dot\relax}%
512 {\dimexpr\xister@Y-#2\xister@five@dot\relax}}%
513 {\xister@five@dot}%
514 \pgfusepath{fill}%
515 \else
516 \pgfsetlinewidth{2\xister@Th}%

```

```

517         \xister@TMPB=\dimexpr\xister@Y-#2\xister@Th\relax
518         \pgfpathmoveto{\pgfpoint{-\xister@X}\xister@TMPB}%
519         \pgfpathlineto
520         {%
521             \pgfpoint
522             {-\dimexpr\xister@X-#3\xister@five@dash\xister@Th*2\relax}%
523             \xister@TMPB
524         }%
525     \fi\fi
526 }%
527 \xister@drawdigit{6#1}%
528 {%
529     \pgfsetlinewidth{2\xister@Th}%
530     \xister@TMPB=\dimexpr\xister@Y-#2\xister@Th\relax
531     \pgfpathmoveto{\pgfpoint{-\xister@X}\xister@TMPB}%
532     \pgfpathlineto{\pgfpoint\xister@XY@share\xister@TMPB}%
533 }%
534 \xister@superimpose{7#1}%
535 {%
536     \rlap{\copy\csname xister@digitbox@1#1\endcsname}%
537     \copy\csname xister@digitbox@6#1\endcsname
538 }%
539 \xister@superimpose{8#1}%
540 {%
541     \rlap{\copy\csname xister@digitbox@2#1\endcsname}%
542     \copy\csname xister@digitbox@6#1\endcsname
543 }%
544 \xister@superimpose{9#1}%
545 {%
546     \rlap{\copy\csname xister@digitbox@1#1\endcsname}%
547     \rlap{\copy\csname xister@digitbox@2#1\endcsname}%
548     \copy\csname xister@digitbox@6#1\endcsname
549 }%
550 }

```

Each digit has the same bounding box and baseline which is specified here. #1 will be the name, #2 the digit specific path.

```

551 \newcommand\xister@drawdigit[2]
552 {%
553     \expandafter\setbox\csname xister@digitbox@#1\endcsname=\hbox
554     {%
555         \begin{pgfpicture}%
556             \pgfpathrectanglecorners
557             {\pgfpoint{-\xister@bx\xister@X}\xister@by\xister@Y}}%
558             {\pgfpoint{\xister@bx\xister@X}\xister@by\xister@Y}}%
559             \pgfusepath{use as bounding box}%
560             #2%
561             \pgfusepath{stroke}%
562             \pgfsetbaseline{\xister@sgn\xister@bs\xister@Y}%
563         \end{pgfpicture}%
564     }%
565 }

```

(End definition for \xister@drawdigits@vertical and \xister@drawdigits@horizontal.)

\xister@drawzero@horizontal
\xister@drawzero@vertical

```

566 \newcommand*\xister@drawzero@vertical
567 {%
568   \xister@drawdigit{0}%
569   {%
570     \pgfsetlinewidth{2\xister@Tv}%
571     \pgfpathmoveto{\pgfpoint\z@{-\xister@Y}}%
572     \pgfpathlineto{\pgfpoint\z@\xister@Y}%
573   }%
574 }
575 \newcommand*\xister@drawzero@horizontal
576 {%
577   \xister@drawdigit{0}%
578   {%
579     \pgfsetlinewidth{2\xister@Th}%
580     \pgfpathmoveto{\pgfpoint{-\xister@X}\z@}%
581     \pgfpathlineto{\pgfpoint\xister@X\z@}%
582   }%
583 }

```

(End definition for \xister@drawzero@horizontal and \xister@drawzero@vertical.)

\xister@superimpose

Some digits can be build from a few of the other ones. This is faster than drawing everything again, and this macro eases the process a bit.

```

584 \newcommand\xister@superimpose[2]
585 {\expandafter\setbox\csname xister@digitbox{#1}\endcsname=\hbox{#2}}

```

(End definition for \xister@superimpose.)

2.6 Rebuild the Glyphs

\xister@setlengths

Lengths are set depending on the current font size and the sizes of \xister@font. To keep the font changes local they are kept inside a group and a chain of \expandafters is used to evaluate the user specified sizes while the \xister@font is still active.

```

586 \protected\def\xister@setlengths
587 {%
588   \begingroup
589   \xister@font
590   \edef\xister@tmp
591   {%
592     \endgroup
593     \xister@X=\the\dimexpr.5\dimexpr\xister@x\relax\relax
594     \xister@Y=\the\dimexpr.5\dimexpr\xister@y\relax\relax
595     \xister@Th=\the\dimexpr.5\dimexpr\xister@th\relax\relax
596     \xister@Tv=\the\dimexpr.5\dimexpr\xister@tv\relax\relax
597     \xister@Tdu=\the\dimexpr.5\dimexpr\xister@tdu\relax\relax
598     \xister@Tdd=\the\dimexpr.5\dimexpr\xister@tdd\relax\relax
599   }%
600   \xister@tmp
601 }

```

(End definition for \xister@setlengths.)

`\xister@ensure@current` The glyphs have to be rebuilt if the font size or colour changed. The latter is an annoyance because we can build boxes in a way that they change colour to the surrounding colour, but PGF is too intelligent here and does issue the corresponding `\specials` to save the current colour for the box. Because of that we test which `\f@size` and colour (using `\\color@.`) was active when we last built the glyphs. If this matches we do nothing, else we rebuild the glyphs.

The `\edef` is just used to remove the debug code if it isn't necessary.

```

602 \protected\edef\xister@ensure@current
603   {%
604     \unexpanded{\edef\xister@tmp{\xister@fontfeatures}}%
605     \unexpanded{\unless\ifx\xister@tmp\xister@last@font}%
606     \ifxister@dbg@
607       \unexpanded
608         {%
609           \typeout
610             {%
611               Package xistercian Debug: Font changed. Building glyphs for
612               \xister@tmp.
613             }%
614         }%
615     \fi
616     \unexpanded
617       {%
618         \let\xister@last@font\xister@tmp
619         \xister@build
620       }%
621     \ifxister@dbg@
622       \unexpanded{\typeout{Package xistercian Debug: Done.}}%
623     \fi
624     \unexpanded{\fi}%
625   }
626 \AtBeginDocument{\xister@ensure@current}

```

(End definition for `\xister@ensure@current`.)

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

A	
<code>\AddToHook</code>	113, 298, 304
<code>\AtBeginDocument</code>	626
B	
<code>\begin</code>	555
<code>\begingroup</code>	102, 169, 588
C	
<code>\cistercian</code>	275
<code>\cisterciannum</code>	276, 277, 286
<code>\cisterciannumE</code>	285
<code>\cistercianredraw</code>	262, 287
<code>\cistercianredrawlazy</code>	263, 287
<code>\cisterciansetup</code>	268, 273
<code>\cistercianstyle</code>	269
<code>\copy</code>	342, 344, 346, 348, 349, 445, 446, 450, 451, 455, 456, 457, 536, 537, 541, 542, 546, 547, 548
cs commands:	
<code>\cs_new_protected:Npn</code>	42
<code>\csname</code>	31, 32, 33, 34, 37, 104, 108, 125, 134, 170, 174, 275, 281, 283, 342, 344, 346, 348, 353, 445, 446, 450, 451, 455, 456, 457, 536, 537, 541, 542, 546, 547, 548, 553, 585
D	
<code>\def</code>	44, 51, 56, 61, 70, 74, 79, 84, 93, 100, 111, 123, 125, 142, 224, 225, 230, 231, 236, 246, 253, 269, 283, 284, 287, 289, 292, 293, 294, 295, 296, 301, 307, 310, 317, 326, 335, 336, 338, 586
<code>\detokenize</code>	104, 106, 108, 319
<code>\dimexpr</code>	243, 257, 357, 364, 370, 374, 381, 387, 397, 408, 415, 420, 421, 425, 432, 439, 462, 466, 473, 479, 489, 499, 506, 511, 512, 517, 522, 530, 593, 594, 595, 596, 597, 598
E	
<code>\edef</code>	46, 48, 127, 147, 148, 149, 150, 152, 154, 155, 156, 157, 160, 166, 183, 196, 203, 277, 590, 602, 604
<code>\ekvcsvloop</code>	188, 189
<code>\ekvdef</code>	144, 192, 199
<code>\ekvdefNoVal</code>	206, 207, 212, 221, 227, 233, 262, 263, 266, 273
<code>\ekvifdefinedNoVal</code>	271
<code>\ekvifdefinedset</code>	140
<code>\ekvlet</code>	218, 219
<code>\ekvletkv</code>	145, 190, 191
<code>\ekvletkvNoVal</code>	264, 265
<code>\ekvoProcessLocalOptions</code>	261
<code>\ekvsetdef</code>	217, 268
<code>\else</code>	107, 173, 315, 418, 424, 509, 515
<code>\end</code>	563
<code>\endcsname</code>	31, 32, 33, 34, 37, 104, 108, 125, 134, 170, 174, 275, 281, 283, 342, 344, 346, 348, 353, 445, 446, 450, 451, 455, 456, 457, 536, 537, 541, 542, 546, 547, 548, 553, 585
<code>\endgroup</code>	102, 169, 592
<code>\endinput</code>	141
exp commands:	
<code>\exp_args:No</code>	117
<code>\expandafter</code>	31, 32, 33, 34, 37, 102, 103, 111, 125, 129, 130, 134, 169, 170, 174, 275, 276, 280, 283, 286, 314, 315, 320, 322, 353, 553, 585
<code>\ExplSyntaxOff</code>	36, 43, 122
<code>\ExplSyntaxOn</code>	28, 41, 112
F	
<code>\fi</code>	109, 175, 302, 308, 315, 323, 357, 364, 434, 525, 615, 623, 624
H	
<code>\hbox</code>	553, 585
I	
<code>\if</code>	319
<code>\ifdefined</code>	300, 306
<code>\ifdim</code>	357, 364
<code>\ifx</code>	103, 170, 314, 605
int commands:	
<code>\int_step_inline:nn</code>	29
L	
<code>\leavevmode</code>	312
<code>\let</code>	38, 39, 40, 53, 54, 72, 76, 77, 95, 97, 98, 99, 161, 162, 163, 174, 182, 188, 220, 223, 229, 297, 618
<code>\long</code>	44, 100, 144, 192, 199, 236, 246, 253, 269

N

`\newcommand` 2, 3, 4, 5, 6, 7, 8, 9,
 10, 11, 12, 13, 14, 15, 275, 276, 285,
 354, 361, 368, 460, 551, 566, 575, 584
`\newdimen` . . . 19, 20, 21, 22, 23, 24, 25, 26, 27
`\newif` 16, 17, 18
`\newsavebox` 31, 32, 33, 34, 37
`\normalfont` 11
`\numexpr` 286, 320

P

`\PackageError` 106, 141, 171, 267, 272
`\pgfpathcircle` 419, 510
`\pgfpathlineto` 376, 383,
 391, 392, 400, 401, 411, 412, 417,
 428, 441, 468, 475, 483, 484, 492,
 493, 502, 503, 508, 519, 532, 572, 581
`\pgfpathmoveto` 375, 382, 390,
 399, 410, 416, 427, 440, 467, 474,
 482, 491, 501, 507, 518, 531, 571, 580
`\pgfpathrectanglecorners` . . 356, 363, 556
`\pgfpoint`
 . 357, 358, 364, 365, 375, 376, 382,
 383, 390, 391, 392, 399, 400, 401,
 410, 411, 412, 416, 417, 420, 427,
 430, 440, 441, 467, 468, 474, 475,
 482, 483, 484, 491, 492, 493, 501,
 502, 503, 507, 508, 511, 518, 521,
 531, 532, 557, 558, 571, 572, 580, 581
`\pgfsetbaseline` 562
`\pgfsetcolor` 53, 54, 72, 76, 77, 95
`\pgfsetlinewidth` 373, 380, 389,
 398, 409, 414, 426, 438, 465, 472,
 481, 490, 500, 505, 516, 529, 570, 579
`\pgfusepath`
 359, 366, 413, 423, 504, 514, 559, 561
`\protected` 44, 51, 74, 100, 125, 144, 192,
 199, 206, 207, 212, 217, 221, 227,
 233, 236, 246, 253, 262, 263, 268,
 269, 273, 283, 287, 292, 338, 586, 602

R

`\relax` 105, 170, 243, 257, 286,
 319, 320, 324, 335, 357, 364, 370,
 374, 381, 387, 397, 408, 415, 420,
 421, 425, 432, 439, 462, 466, 473,
 479, 489, 499, 506, 511, 512, 517,
 522, 530, 593, 594, 595, 596, 597, 598
`\RequirePackage` 1
`\rlap` 295, 342, 344, 346, 348,
 445, 450, 455, 456, 536, 541, 546, 547

S

`\setbox` 553, 585

str commands:
`\str_if_in:nnTF` 117

T

T_EX and L^AT_EX 2_ε commands:
`\@backslashchar` 134
`\@cistercian` 275
`\@empty` 182, 188
`\@firstofone` 324
`\@firstoftwo` 297
`\@gobble` 38, 39, 40, 54, 77
`\@gobbletwo` 322
`\@ifpackageloaded` 115
`\cisterciannum@pdf` 277
`\f@encoding` 135
`\f@family` 136
`\f@series` 137
`\f@shape` 138
`\f@size` 111, 139
`\ifxister@dbg@` 16, 606, 621
`\ifxister@five@dot@` 18, 418, 509
`\ifxister@five@triangle@` . . 17, 405, 497
`\xister@a` 283, 310
`\xister@add@fontfeature` 100, 188, 189
`\xister@add@fontfeature@encoding` 123
`\xister@add@fontfeature@family` . 123
`\xister@add@fontfeature@series` . 123
`\xister@add@fontfeature@shape` . . 123
`\xister@add@fontfeature@size` . . . 123
`\xister@b` 310
`\xister@bs` 10, 157, 562
`\xister@build` . . . 51, 174, 223, 229, 619
`\xister@build@h` 51
`\xister@build@horizontal` . . . 51, 223
`\xister@build@v` 51
`\xister@build@vertical` 51, 229
`\xister@bx` 8, 155, 557, 558
`\xister@by` 9, 156, 557, 558
`\xister@c` 310
`\xister@clip@h` 354, 480, 488, 498
`\xister@clip@v` 354, 388, 396, 406
`\xister@d` 310
`\xister@dbg@dim` 38, 147,
 148, 149, 150, 152, 154, 164, 197, 236
`\xister@dbg@div` 39, 49, 253
`\xister@dbg@float`
 . . 38, 47, 155, 156, 157, 184, 204, 246
`\xister@dbg@true` 235
`\xister@done` 293, 340

<code>\xister@drawdigit</code>	371, 378, 385, 394, 403, 436, 463, 470, 477, 486, 495, 527, 551, 568, 577
<code>\xister@drawdigits@horizontal</code> ..	81, 85, 90, 94, 368
<code>\xister@drawdigits@vertical</code>	58, 62, 67, 71, 368
<code>\xister@drawzero@horizontal</code> .	80, 566
<code>\xister@drawzero@vertical</code> ..	57, 566
<code>\xister@ensure@current</code> .	290, 313, 602
<code>\xister@five@dash</code> ..	13, 203, 432, 522
<code>\xister@five@dot</code>	12, 196, 420, 421, 422, 511, 512, 513
<code>\xister@five@dot@false</code>	202, 215
<code>\xister@five@dot@true</code>	195, 210
<code>\xister@five@set</code> ..	217, 218, 219, 220
<code>\xister@five@triangle@false</code>	194, 201, 209, 214
<code>\xister@five@triangle@true</code> ..	17, 206
<code>\xister@font</code>	11, 166, 589
<code>\xister@fontfeatures</code>	100, 127, 130, 188, 604
<code>\xister@fraction</code>	44, 180
<code>\xister@gobblecopy</code>	293, 341, 343, 345, 347
<code>\xister@gobbletoat</code>	293, 328, 340
<code>\xister@gobbletozero</code>	293, 320, 341, 343, 345, 347
<code>\xister@if@slash</code>	41, 179
<code>\xister@last@font</code> .	289, 292, 605, 618
<code>\xister@pgfsetcolor</code>	53, 72, 76, 95
<code>\xister@setlengths</code>	55, 78, 586
<code>\xister@sgn</code>	56, 61, 70, 79, 84, 93, 374, 415, 562
<code>\xister@share@div</code> .	14, 48, 182, 370, 462
<code>\xister@share@mul</code> .	15, 46, 183, 370, 462
<code>\xister@stop</code>	44, 180
<code>\xister@superimpose</code>	443, 448, 453, 534, 539, 544, 584
<code>\xister@symbols</code> ...	296, 321, 335, 336
<code>\xister@Tdd</code>	25, 63, 64, 86, 87, 387, 389, 489, 490, 499, 500, 598
<code>\xister@tdd</code>	5, 154, 163, 598
<code>\xister@Tdu</code>	24, 64, 65, 87, 88, 397, 398, 408, 409, 479, 481, 597
<code>\xister@tdu</code>	4, 152, 162, 597
<code>\xister@texorpdf</code>	279, 297
<code>\xister@Th</code>	22, 364, 373, 374, 380, 381, 414, 415, 479, 489, 499, 516, 517, 522, 529, 530, 579, 595
<code>\xister@th</code>	2, 150, 161, 595
<code>\xister@tmp</code>	123, 134, 135, 136, 137, 138, 139, 142, 147, 148, 149, 150, 151, 153, 155, 156, 157, 158, 166, 167, 177, 187, 189, 590, 600, 604, 605, 612, 618
<code>\xister@TMPA</code>	26, 63, 65, 86, 88, 243, 250, 257, 374, 375, 376, 381, 382, 383, 415, 416, 417, 466, 467, 468, 473, 474, 475, 506, 507, 508
<code>\xister@TMPB</code>	27, 387, 390, 391, 397, 399, 400, 407, 410, 411, 425, 427, 431, 439, 440, 441, 479, 482, 483, 489, 491, 492, 499, 501, 502, 517, 518, 523, 530, 531, 532
<code>\xister@Tv</code>	23, 357, 387, 397, 408, 425, 426, 432, 438, 439, 465, 466, 472, 473, 505, 506, 570, 596
<code>\xister@tv</code> .	3, 149, 160, 161, 162, 163, 596
<code>\xister@undefined</code>	220
<code>\xister@X</code> .	19, 59, 66, 68, 82, 89, 91, 357, 358, 365, 376, 383, 392, 401, 412, 417, 420, 425, 439, 462, 466, 483, 491, 493, 501, 503, 506, 511, 518, 522, 531, 557, 558, 580, 581, 593
<code>\xister@x</code>	6, 147, 224, 230, 593
<code>\xister@XY@share</code>	21, 370, 381, 388, 390, 392, 400, 411, 441, 462, 473, 480, 482, 484, 492, 502, 532
<code>\xister@Y</code> ..	20, 60, 69, 83, 92, 358, 364, 365, 370, 374, 391, 399, 401, 410, 412, 415, 421, 427, 432, 440, 468, 475, 484, 493, 503, 508, 512, 517, 530, 557, 558, 562, 571, 572, 594
<code>\xister@y</code>	7, 148, 225, 231, 594
<code>\z@</code>	250, 257, 357, 364, 375, 382, 396, 406, 416, 467, 474, 488, 498, 507, 571, 572, 580, 581
<code>\texorpdfstring</code>	300, 301, 306, 307
<code>\the</code>	276, 286, 320, 593, 594, 595, 596, 597, 598
tl commands:	
<code>\tl_if_in:nnTF</code>	42
<code>\typeout</code>	238, 244, 248, 251, 255, 258, 609, 622
U	
<code>\unexpanded</code>	46, 48, 111, 129, 130, 147, 148, 149, 150, 152, 154, 155, 156, 157, 160, 166, 183, 196, 203, 241, 249, 256, 279, 280, 604, 605, 607, 616, 622, 624
<code>\unless</code>	609