

The xistercian package

Jonathan P. Spratte*

2025-04-13 v1.3
f - y - t

Contents

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

1 Documentation

The xistercian package provides Cistercian numerals for use in L^AT_EX. The name is chosen to be xistercian because of the older (but earlier unreleased) [cistercian package](#), which has the same goal but goes a different way implementing it. In that other package you can change the way things look using TikZ keys, so you might want to check it out and see which of the two fits your needs best.

*jspratte@yahoo.de

Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ	Ⅶ	Ⅷ	Ⅸ
1	2	3	4	5	6	7	8	9
Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ	Ⅶ	Ⅷ	Ⅸ
10	20	30	40	50	60	70	80	90
Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ	Ⅶ	Ⅷ	Ⅸ
100	200	300	400	500	600	700	800	900
Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ	Ⅶ	Ⅷ	Ⅸ
1000	2000	3000	4000	5000	6000	7000	8000	9000

Figure 1: The different base glyphs in vertical mode

Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ	Ⅶ	Ⅷ	Ⅸ
1	2	3	4	5	6	7	8	9
Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ	Ⅶ	Ⅷ	Ⅸ
10	20	30	40	50	60	70	80	90
Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ	Ⅶ	Ⅷ	Ⅸ
100	200	300	400	500	600	700	800	900
Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ	Ⅶ	Ⅷ	Ⅸ
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 (Ⅰ) 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 (Ⅰ), upper left tens (Ⅱ), lower right hundreds (Ⅲ), and lower left thousands (Ⅳ). 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.

Moreover some glyphs have a different shape in some manuscripts. The shapes in figures 1 and 2 are the most used ones nowadays (most likely because those are the ones Wikipedia shows vector graphics of). This package offers some (not all) alternative forms. Most notably some historic documents have swapped shapes of digits. To not further the confusion this package doesn't support swapping digit shapes.

All glyphs have the same bounding box, are drawn using the pgf package, and cached inside T_EX 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 (to my knowledge there are no Unicode points for Cistercian numerals).

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 -𐌹𐌺. And in the case that four consecutive digits are zeros they get displayed as that: 𐌹𐌺 is 10000.

Full disclaimer: I'm neither a historian nor have I any authority on the matter. I'm just a bloke who was fascinated by this numeral system and wanted to provide L^AT_EX support for it. Most of my knowledge about this stems from Wikipedia and similar sources.

1.2 Macros

```
\cistercian \cistercian{<counter>}
```

Prints the value of the L^AT_EX 2_ε *<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.*

```
\cisterciannum \cisterciannum{<integer>}
```

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.

```
\cistercianeval \cistercianeval{<integer expression>}
```

Evaluates the *<integer expression>* and prints the result as a Cistercian numeral.

```
\cisterciansetup \cisterciansetup{<key=value,...>}
```

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.

```
\cistercianredraw \cistercianredraw
```

```
\cistercianredrawlazy
```

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.

```
\cistercianstyle \cistercianstyle{*}{<name>}{<key=value,...>}
```

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. If your style shouldn't take an argument (so is fixed) use the normal variant. If your style needs to take an argument use the variant with *<*>*.

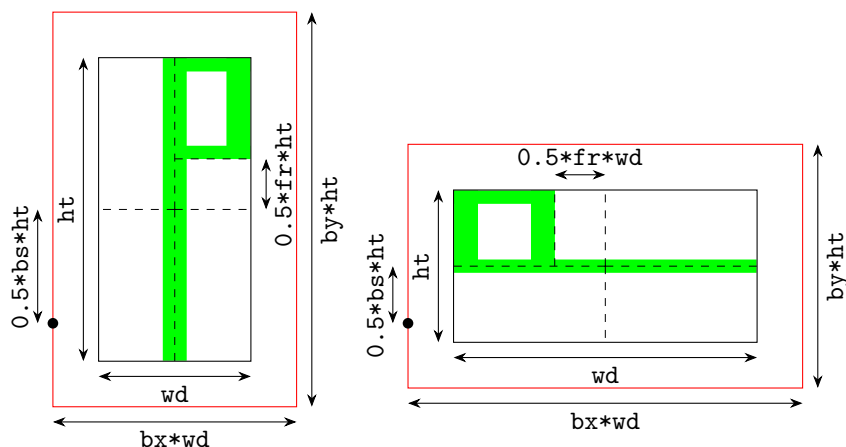


Figure 3: Measurements of a glyph. The red rectangle is the bounding box, the dot on the left shows the reference point placed on the surrounding baseline.

For example, the key `horizontal` is equivalently defined to and can be used as

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

And if you want to define a style that at the same time changes the stroke widths for vertical and horizontal strokes you could use

```
\cistercianstyle*{sh+v}{sh={#1},sv={#1}}
\cisterciansetup{sh+v=.12ex}
```

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. A *<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.

<code>width</code>	<code>width = {<dimension expression>}</code>	<code>.5em</code>
<code>wd</code>	Set the width of the Cistercian numeral digits.	
<code>height</code>	<code>height = {<dimension expression>}</code>	<code>1.55ex</code>
<code>ht</code>	Set the height of the Cistercian numeral digits.	

[†]`<key>=<value>` interface powered by `expkv`

<hr/>	<code>bound-x</code>	<code>bound-x = {<float>}</code>	1.2
<hr/>	<code>bx</code>	Specifies the factor the bounding box in x -direction is bigger than the width. An empty $\langle float \rangle$ is considered the same as 1.	
<hr/>	<code>bound-y</code>	<code>bound-y = {<float>}</code>	<i>empty</i>
<hr/>	<code>by</code>	Specifies the factor the bounding box in y -direction is bigger than the height. An empty $\langle float \rangle$ is considered the same as 1.	
<hr/>	<code>baseline</code>	<code>baseline = {<float>}</code>	<i>empty</i>
<hr/>	<code>bs</code>	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.	
<hr/>	<code>stroke-v</code>	<code>stroke-v = {<dimension expression>}</code>	.18ex
<hr/>	<code>sv</code>	Sets the stroke thickness of vertical strokes.	
<hr/>	<code>stroke-h</code>	<code>stroke-h = {<dimension expression>}</code>	.1ex
<hr/>	<code>sh</code>	Sets the stroke thickness of horizontal strokes.	
<hr/>	<code>stroke-du</code>	<code>stroke-du = {<dimension expression>}</code>	.1ex
<hr/>	<code>sdu</code>	Sets the stroke thickness of diagonal strokes going from lower left to upper right.	
<hr/>	<code>stroke-dd</code>	<code>stroke-dd = {<dimension expression>}</code>	.18ex
<hr/>	<code>sdd</code>	Sets the stroke thickness of diagonal strokes going from upper left to lower right.	
<hr/>	<code>stroke-s</code>	<code>stroke-s = {<dimension expression>}</code>	<i>empty</i>
<hr/>	<code>ss</code>	Sets the stroke thickness of the zero stem. If this is set to an empty value the stroke thickness in vertical mode is that of <code>stroke-v</code> and in horizontal mode that of <code>stroke-h</code> .	
<hr/>	<code>strokes</code>	<code>stroke = {<dimension expression>}</code>	
<hr/>	<code>s</code>	Sets all stroke thicknesses at once.	
<hr/>	<code>fraction</code>	<code>fraction = {<float>}</code>	/6
<hr/>	<code>fr</code>	<code>fraction = {<float>}/{<integer>}</code>	
<hr/>		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 <code>width</code> , and the upper border by <code>height</code> , 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 $\langle integer \rangle$ (with a leading slash), or both. An empty $\langle float \rangle$ equals 1.	
		Consider the following example (result on the right):	
		<code>{\cisterciansetup{fr=2/9,redraw}\cisterciannum{9}}\par</code>	¶
		<code>{\cisterciansetup{fr=0.6,redraw}\cisterciannum{9}}\par</code>	¶

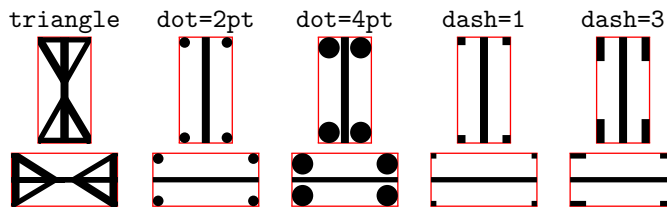


Figure 4: Alternate forms of the digit 5. The length of the dash=3 variant is too long in the normal setup of this document, as the difference between `||` and `!!!` is hard to notice. The tight bounding boxes using `bx=1,by=1` are drawn red.

```
orientation orientation = {<choice>} vertical
```

^o 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.

```
horizontal horizontal
```

Same as setting `orientation = horizontal`, `width = 0.775em`, `height = 1.13ex`.

```
vertical vertical
```

Same as setting `orientation = vertical`, `width = 0.5em`, `height = 1.55ex`.

```
alternate-5 alternate-5 = {<choice>} triangle
```

⁵ 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*. Please ensure that your readers can't confuse the results with the shape of 6.

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

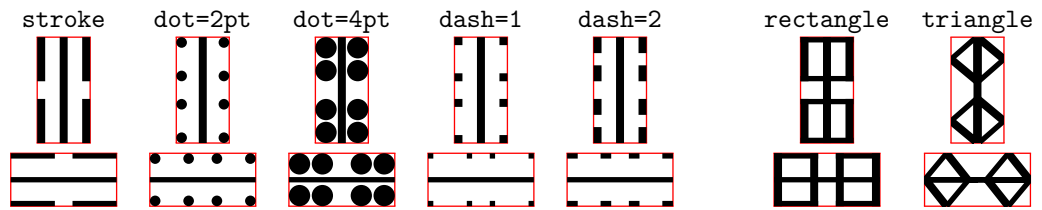


Figure 5: Alternate forms of the digits 6 (left) and 9 (right). The tight bounding boxes using `bx=1,by=1` are drawn red.

```
alternate-6 \alternate-6 = {<choice>} stroke
```

6 There are alternative forms of the digit 6. While today the usual form is `stroke`, there are also historic documents showing the 6 as two dots, which you can achieve using the `dot` choice, and to support squares or short dashes instead of round dots you can use the `dash` choice.

For `dot` you can also specify the radius as a dimension. The default is `.09ex`. And for `dash` you can also specify the length in multiples of the used `stroke` thickness, an empty value is considered the same as `1`. The default is `empty`. The `stroke` has always the full length and doesn't accept an additional specification.

A comparison of the alternate forms is shown in figure 5.

```
alternate-9 \alternate-9 = {<choice>} rectangle
```

9 Another digit having alternative forms is the 9. For this two `<choice>`s are available, `rectangle` and `triangle`. Neither of the two accept additional specifications.

A comparison of the alternate forms is shown in figure 5.

```
font font = {<font setup>} \normalfont
```

f 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`).

```

redraw redraw = {<feature-list>} size
r      redraw += {<feature-list>}

```

redraw redraw!

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.

```

debug debug

```

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 say who we are and load the required packages

```
1 \ProvidesPackage{xistercian}[2025-04-13 v1.3 Cistercian numerals in LaTeX]
2 \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.

```
3 \newcommand*\xister@th{.1ex}
4 \newcommand*\xister@tv{.18ex}
5 \newcommand*\xister@tdu{.1ex}
6 \newcommand*\xister@tdd{.18ex}
7 \newcommand*\xister@ts{}
8 \newcommand*\xister@x{.5em}
9 \newcommand*\xister@y{1.55ex}
10 \newcommand*\xister@bx{1.2}
11 \newcommand*\xister@by{}
12 \newcommand*\xister@bs{}
13 \newcommand*\xister@font{\normalfont}
14 \newcommand*\xister@five@dot{.09ex}
15 \newcommand*\xister@five@dash{}
16 \newcommand*\xister@six@dot{.09ex}
17 \newcommand*\xister@six@dash{}
18 \newcommand*\xister@share@div{/6}
19 \newcommand*\xister@share@mul{}
20 \newif\ifxister@dbg@
21 \newif\ifxister@five@triangle@\xister@five@triangle@true
22 \newif\ifxister@five@dot@
23 \newif\ifxister@six@dot@
24 \newif\ifxister@six@stroke@\xister@six@stroke@true
25 \newif\ifxister@nine@triangle@
26 \newdimen\xister@X
27 \newdimen\xister@Y
28 \newdimen\xister@XY@share
29 \newdimen\xister@Th
30 \newdimen\xister@Tv
31 \newdimen\xister@Tdu
32 \newdimen\xister@Tdd
33 \newdimen\xister@Ts
34 \newdimen\xister@TMPA
35 \newdimen\xister@TMPB
36 \newdimen\xister@Five@dot
37 \newdimen\xister@Six@dot
```

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

```
38 \ExplSyntaxOn
39 \int_step_inline:nn {9}
40 {
41   \expandafter\newsavebox\csgname xister@digitbox@#1\endcsgname
42   \expandafter\newsavebox\csgname xister@digitbox@#10\endcsgname
```

```

43 \expandafter\newsavebox\csname xister@digitbox@#100\endcsname
44 \expandafter\newsavebox\csname xister@digitbox@#1000\endcsname
45 }
46 \ExplSyntaxOff
47 \expandafter\newsavebox\csname xister@digitbox@0\endcsname

```

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

```

48 \let\xister@dbg@dim\@gobble
49 \let\xister@dbg@div\@gobble
50 \let\xister@dbg@float\@gobble

```

(End of 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.

```

51 \ExplSyntaxOn
52 \cs_new_protected:Npn \xister@if@slash #1 { \tl_if_in:nnTF {#1} { / } }
53 \ExplSyntaxOff

```

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

`\xister@fraction`

```

54 \protected\long\def\xister@fraction#1/#2\xister@stop
55 {%
56 \edef\xister@share@mul{\unexpanded{#1}}%
57 \xister@dbg@float{#1}%
58 \edef\xister@share@div{/\unexpanded{#2}}%
59 \xister@dbg@div{#2}%
60 }

```

(End of 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 the
`\xister@build@v` correct direction. We locally disable `\pgfsetcolor` to allow building colour-unsafe boxes.
`\xister@build@vertical` This only works as long as `luacolor` (or something equivalent) isn't used, so consider this
a crude hack.

```

61 \protected\def\xister@build@vertical
62 {%
63 \let\xister@pgfsetcolor\pgfsetcolor
64 \let\pgfsetcolor\@gobble
65 \xister@setlengths
66 \def\xister@sgn{-}%
67 \xister@drawzero@vertical
68 \xister@drawdigits@vertical{}{}{}%
69 \xister@X=-\xister@X
70 \xister@Y=-\xister@Y
71 \def\xister@sgn{+}%

```

```

72 \xister@drawdigits@vertical{000}{-}{-}%
73 \xister@TMPA=\xister@Tdd
74 \xister@Tdd=\xister@Tdu
75 \xister@Tdu=\xister@TMPA
76 \xister@X=-\xister@X
77 \xister@drawdigits@vertical{00}{-}{-}%
78 \xister@X=-\xister@X
79 \xister@Y=-\xister@Y
80 \def\xister@sgn{-}%
81 \xister@drawdigits@vertical{0}{-}{-}%
82 \let\pgfsetcolor\xister@pgfsetcolor
83 }
84 \protected\def\xister@build@horizontal
85 {%
86 \let\xister@pgfsetcolor\pgfsetcolor
87 \let\pgfsetcolor@gobble
88 \xister@setlengths
89 \def\xister@sgn{-}%
90 \xister@drawzero@horizontal
91 \xister@drawdigits@horizontal{}{}{}%
92 \xister@X=-\xister@X
93 \xister@Y=-\xister@Y
94 \def\xister@sgn{+}%
95 \xister@drawdigits@horizontal{000}{-}{-}%
96 \xister@TMPA=\xister@Tdd
97 \xister@Tdd=\xister@Tdu
98 \xister@Tdu=\xister@TMPA
99 \xister@X=-\xister@X
100 \xister@drawdigits@horizontal{0}{-}{-}%
101 \xister@X=-\xister@X
102 \xister@Y=-\xister@Y
103 \def\xister@sgn{-}%
104 \xister@drawdigits@horizontal{00}{-}{-}%
105 \let\pgfsetcolor\xister@pgfsetcolor
106 }
107 \let\xister@build\xister@build@vertical
108 \let\xister@build@v\xister@build@vertical
109 \let\xister@build@h\xister@build@horizontal

```

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

\xister@fontfeatures 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.

\xister@add@fontfeature

```

110 \protected\long\def\xister@add@fontfeature#1%
111 {%
112 \begingroup\expandafter\expandafter\expandafter\endgroup
113 \expandafter\ifx
114 \csname xister@add@fontfeature@\detokenize{#1}\endcsname
115 \relax
116 \PackageError{xistercian}{Unsupported font feature '\detokenize{#1}'.}{-}%
117 \else
118 \csname xister@add@fontfeature@\detokenize{#1}\endcsname
119 \fi
120 }
121 \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.

```

122 \ExplSyntaxOn
123 \AddToHook{begindocument/before}
124 {
125   \@ifpackageloaded { luacolor }
126   {
127     \exp_args:No \str_if_in:nmF \xister@fontfeatures { color }
128     { \xister@add@fontfeature{color} }
129   }
130   {}
131 }
132 \ExplSyntaxOff

```

(End of 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.

```

133 \def\xister@tmp#1#2%
134 {%
135   \protected\expandafter\def\csname xister@add@fontfeature@#2\endcsname
136   {%
137     \edef\xister@fontfeatures
138     {%
139       \unexpanded\expandafter
140       {\xister@fontfeatures\unexpanded\expandafter{#1}}/%
141     }%
142   }%
143 }
144 \expandafter\xister@tmp\csname \@backslashchar color@.\endcsname{color}
145 \xister@tmp\f@encoding{encoding}
146 \xister@tmp\f@family {family}
147 \xister@tmp\f@series {series}
148 \xister@tmp\f@shape {shape}
149 \xister@tmp\f@size {size}

```

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

Now we define the keys

```

150 \ekvifdefinedset{xister}
151 {\PackageError{xistercian}{keyval conflict detected. Aborting}{ }\endinput}{ }

```

We'll define a shorthand name and a long name for all the keys, so we set up an auxiliary here for these definitions that'll just take two key names instead of one for each definition.

```

152 \def\xister@tmp#1#2#3%
153 {%
154   \protected\long\ekvdef{xister}{#1}{#3}%
155   \ekvletkv{xister}{#2}{xister}{#1}%
156 }
157 \xister@tmp{wd}{width}{\edef\xister@x{\unexpanded{#1}}\xister@dbg@dim{#1}}
158 \xister@tmp{ht}{height}{\edef\xister@y{\unexpanded{#1}}\xister@dbg@dim{#1}}
159 \xister@tmp{bx}{bound-x}{\edef\xister@bx{\unexpanded{#1}}\xister@dbg@float{#1}}
160 \xister@tmp{by}{bound-y}{\edef\xister@by{\unexpanded{#1}}\xister@dbg@float{#1}}
161 \xister@tmp{bs}{baseline}{\edef\xister@bs{\unexpanded{#1}}\xister@dbg@float{#1}}

```

```

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

```

```

216   {%
217     \xister@five@triangle@false
218     \xister@five@dot@false
219     \edef\xister@five@dash{\unexpanded{#1}}%
220     \xister@dbg@float{#1}%
221   }
222 \protected\ekvdefNoVal{xister/5}{triangle}{\xister@five@triangle@true}
223 \protected\ekvdefNoVal{xister/5}{dot}
224   {%
225     \xister@five@triangle@false
226     \xister@five@dot@true
227   }
228 \protected\ekvdefNoVal{xister/5}{dash}
229   {%
230     \xister@five@triangle@false
231     \xister@five@dot@false
232   }

```

Starting here the `\xister@tmp` macro is no longer the shorthand for the doubled definitions!

```

233 \protected\ekvsetdef\xister@tmp{xister/5}
234 \ekvlet{xister}{5}\xister@tmp
235 \ekvlet{xister}{alternate-5}\xister@tmp
236 \protected\long\ekvdef{xister/6}{dot}
237   {%
238     \xister@six@stroke@false
239     \xister@six@dot@true
240     \edef\xister@six@dot{\unexpanded{#1}}%
241     \xister@dbg@dim{#1}%
242   }
243 \protected\long\ekvdef{xister/6}{dash}
244   {%
245     \xister@six@stroke@false
246     \xister@six@dot@false
247     \edef\xister@six@dash{\unexpanded{#1}}%
248     \xister@dbg@float{#1}%
249   }
250 \protected\ekvdefNoVal{xister/6}{stroke}{\xister@six@stroke@true}
251 \protected\ekvdefNoVal{xister/6}{dot}
252   {%
253     \xister@six@stroke@false
254     \xister@six@dot@true
255   }
256 \protected\ekvdefNoVal{xister/6}{dash}
257   {%
258     \xister@six@stroke@false
259     \xister@six@dot@false
260   }
261 \protected\ekvsetdef\xister@tmp{xister/6}
262 \ekvlet{xister}{6}\xister@tmp
263 \ekvlet{xister}{alternate-6}\xister@tmp
264 \protected\ekvdefNoVal{xister/9}{rectangle}{\xister@nine@triangle@false}
265 \protected\ekvdefNoVal{xister/9}{triangle}{\xister@nine@triangle@true}
266 \protected\ekvsetdef\xister@tmp{xister/9}

```

```

267 \ekvlet{xister}{9}\xister@tmp
268 \ekvlet{xister}{alternate-9}\xister@tmp
269 \protected\ekvdefNoVal{xister}{horizontal}
270   {%
271     \let\xister@build\xister@build@horizontal
272     \def\xister@x{.775em}%
273     \def\xister@y{1.13ex}%
274   }
275 \protected\ekvdefNoVal{xister}{vertical}
276   {%
277     \let\xister@build\xister@build@vertical
278     \def\xister@x{.5em}%
279     \def\xister@y{1.55ex}%
280   }

```

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

```

281 \protected\ekvdefNoVal{xister}{debug}
282   {%
283     \xister@dbg@true
284     \protected\long\def\xister@dbg@msg##1%
285       {%
286         \begingroup
287           \newlinechar`^^J
288           \message{##1}%
289         \endgroup
290       }%
291     \protected\long\def\xister@dbg@do##1%
292       {\xister@dbg@msg{^^JPackage xistercian Debug: ##1...}}%
293     \protected\long\def\xister@dbg@done{\xister@dbg@msg{ Done.^^J}}%
294     \protected\long\def\xister@dbg@dim##1%
295       {%
296         \xister@dbg@do{Trying to use dimension ‘\unexpanded{##1}’}%
297         \xister@TMPA=\dimexpr##1\relax
298         \xister@dbg@done
299       }%
300     \protected\long\def\xister@dbg@float##1%
301       {%
302         \xister@dbg@do{Trying to use float ‘\unexpanded{##1}’}%
303         \xister@TMPA=##1\z@
304         \xister@dbg@done
305       }%
306     \protected\long\def\xister@dbg@div##1%
307       {%
308         \xister@dbg@do{Trying to use divisor ‘\unexpanded{##1}’}%
309         \xister@TMPA=\dimexpr\z@/##1\relax
310         \xister@dbg@done
311       }%
312   }

```

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).

```

313 \ekvoProcessLocalOptions{xister}

```

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

```

314 \protected\ekvdefNoVal{xister}{r!}{\cistercianredraw}

```



```

315 \protected\ekvdefNoVal{xister}{r} {\cistercianredrawlazy}
316 \ekvletkvNoVal{xister}{redraw} {xister}{r}
317 \ekvletkvNoVal{xister}{redraw!}{xister}{r!}
    The debug option is only available as a package option, we let it throw an error:
318 \ekvdefNoVal{xister}{debug}
319 {\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.

```
320 \protected\ekvsetdef\cisterciansetup{xister}
```

(End of 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 `\xister@style`. 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`, or a normal key if their style needs to take a value.

```

321 \protected\def\cistercianstyle
322   {%
323     \@ifstar
324     {\xister@style\ekvifdefined{\long\ekvdef}}%
325     {\xister@style\ekvifdefinedNoVal\ekvdefNoVal}%
326   }
327 \protected\long\def\xister@style#1#2#3#4%
328   {%
329     #1{xister}{#3}%
330     {\PackageError{xistercian}{Key ‘#3’ already defined}{}}%
331     {\protected#2{xister}{#3}{\cisterciansetup{#4}}}%
332   }

```

(End of definition for \cistercianstyle and \xister@style. These functions are 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`.

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

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

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

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

\cisterciannum
`\cisterciannum@pdf`

```

335 \edef\cisterciannum
336   {%
337     \unexpanded{\xister@texorpdf}%
338     \unexpanded\expandafter
339     {\\csname cisterciannum \endcsname\cisterciannum@pdf}%
340   }
341 \protected\expandafter\def\csname cisterciannum \endcsname#1{\xister@a#1@}
342 \def\cisterciannum@pdf#1{CISTER#1}

```

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

`\cistercianeval` The second macro is just there for backwards compatibility and deprecated.

```
\cisterciannumE 343 \newcommand\cistercianeval[1]
344   {\expandafter\cisterciannum\expandafter{\the\numexpr#1\relax}}
345 \let\cisterciannumE\cistercianeval
```

(End of definition for \cistercianeval and \cisterciannumE. These functions are documented on page 1.)

`\cistercianredraw`
`\cistercianredrawlazy`

```
346 \protected\def\cistercianredraw
347   {%
348   \def\xister@last@font{\xister@last@font}%
349   \xister@ensure@current
350   }
351 \protected\def\cistercianredrawlazy{\def\xister@last@font{\xister@last@font}}
```

(End of 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 352 \def\xister@gobbletoat#1{}
\xister@gobblecopy 353 \def\xister@gobbletozero#10{}
\xister@done 354 \def\xister@gobblecopy0\rlap#1{}
355 \def\xister@done#1\xister@symbols{}
```

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

`\xister@texorpdf`

```
356 \let\xister@texorpdf\@firstoftwo
357 \AddToHook{package/hyperref/after}
358   {%
359   \ifdefined\texorpdfstring
360   \def\xister@texorpdf{\texorpdfstring}%
361   \fi
362   }
363 \AddToHook{begindocument/before}
364   {%
365   \ifdefined\texorpdfstring
366   \def\xister@texorpdf{\texorpdfstring}%
367   \fi
368   }
```

(End of 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 to
`\xister@d` step over a minus sign and go to the next step.

```
369 \def\xister@a#1%
370   {%
371     \leavevmode
372     \xister@ensure@current
373     \ifx-#1-\expandafter\xister@b
374     \else\expandafter\xister@b\expandafter#1\fi
375   }
```

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 `@s` serve as end markers since each step of both `\xister@c` grabs eight and each of `\xister@symbols` will grab four digits. The `\xister@c` is used to put enough zeros in front of our number such that it consists of blocks of four digits.

```
376 \def\xister@b#1%
377   {%
378     \if\relax\detokenize{#1}\relax
379     \expandafter\xister@gobbletozero\the\numexpr\relax
380     \xister@symbols0000@@@%
381     \expandafter\@gobbletwo
382     \fi
383     \@firstofone{\xister@c#1@@@@@@\relax#1@@@@}%
384   }
385 \def\xister@c#1#2#3#4#5#6#7#8%
386   {%
387     \xister@gobbletoat
388     #1\xister@d{}%
389     #2\xister@d{00}%
390     #3\xister@d{00}%
391     #4\xister@d0%
392     #5\xister@d{}%
393     #6\xister@d{000}%
394     #7\xister@d{00}%
395     #8\xister@d0%
396     @\xister@c
397   }
398 \def\xister@d#1#2@\xister@c#3\relax{\xister@symbols#1}
```

(End of 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.

```
399 \def\xister@symbols#1%
400   {%
401     \protected\def\xister@symbols##1##2##3##4%
402     {%
```

```

403     \xister@gobbletoat##1\xister@done @%
404     \xister@gobbletozero##1\xister@gobblecopy0%
405         \rlap{\copy\csname xister@digitbox@##1000\endcsname}%
406     \xister@gobbletozero##2\xister@gobblecopy0%
407         \rlap{\copy\csname xister@digitbox@##200\endcsname}%
408     \xister@gobbletozero##3\xister@gobblecopy0%
409         \rlap{\copy\csname xister@digitbox@##30\endcsname}%
410     \xister@gobbletozero##4\xister@gobblecopy0%
411         \rlap{\copy\csname xister@digitbox@##4\endcsname}%
412     \copy#1%
413     \xister@symbols
414 }%
415 }
416 \expandafter\xister@symbols\csname xister@digitbox@0\endcsname

```

(End of 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
`\xister@clip@h` to select a clipping region.

```

417 \newcommand*\xister@clip@v[1]
418 {%
419     \pgfpathrectanglecorners
420     {\pgfpoint{\dimexpr\ifdim\xister@X<\z@-\fi\xister@Ts\relax}{#1}}%
421     {\pgfpoint\xister@X\xister@Y}%
422     \pgfusepath{clip}%
423 }
424 \newcommand*\xister@clip@h[1]
425 {%
426     \pgfpathrectanglecorners
427     {\pgfpoint{#1}{-\dimexpr\ifdim\xister@Y<\z@-\fi\xister@Ts\relax}}%
428     {\pgfpoint{-\xister@X}\xister@Y}%
429     \pgfusepath{clip}%
430 }

```

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

`\xister@drawdigits@vertical` The digit glyphs. Since each numeral is build from the zero stem and some appendix to
`\xister@drawdigits@horizontal` 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.

```

431 \newcommand\xister@drawdigits@vertical [3]
432 {%
433     \xister@XY@share=\dimexpr\xister@share@mul\xister@Y\xister@share@div\relax
434     \xister@drawdigit{1#1}%
435     {%
436         \pgfsetlinewidth{2\xister@Th}%
437         \xister@TMPA=\dimexpr\xister@Y\xister@sgn\xister@Th\relax
438         \pgfpathmoveto{\pgfpoint\z@\xister@TMPA}%
439         \pgfpathlineto{\pgfpoint\xister@X\xister@TMPA}%
440     }%
441     \xister@drawdigit{2#1}%
442     {%
443         \pgfsetlinewidth{2\xister@Th}%

```

```

444     \xister@TMPA=\dimexpr\xister@XY@share+#2\xister@Th\relax
445     \pgfpathmoveto{\pgfpoint\z@\xister@TMPA}%
446     \pgfpathlineto{\pgfpoint\xister@X\xister@TMPA}%
447   }%
448 \xister@drawdigit{3#1}%
449   {%
450     \xister@TMPB=#3\dimexpr\xister@Ts-\xister@Tdd\relax
451     \xister@clip@v\xister@XY@share
452     \pgfsetlinewidth{2\xister@Tdd}%
453     \pgfpathmoveto{\pgfpoint\xister@TMPB\xister@XY@share}%
454     \pgfpathlineto{\pgfpoint\xister@TMPB\xister@Y}%
455     \pgfpathlineto{\pgfpoint\xister@X\xister@XY@share}%
456   }%
457 \xister@drawdigit{4#1}%
458   {%
459     \xister@clip@v\z@
460     \xister@TMPB=#3\dimexpr\xister@Ts-\xister@Tdu\relax
461     \pgfsetlinewidth{2\xister@Tdu}%
462     \pgfpathmoveto{\pgfpoint\xister@TMPB\xister@Y}%
463     \pgfpathlineto{\pgfpoint\xister@TMPB\xister@XY@share}%
464     \pgfpathlineto{\pgfpoint\xister@X\xister@Y}%
465   }%
466 \xister@drawdigit{5#1}%
467   {%
468     \ifxister@five@triangle@
469       \xister@clip@v\z@
470       \xister@TMPB=%
471         #3\dimexpr\xister@Ts-\xister@Tdu\relax
472       \pgfsetlinewidth{2\xister@Tdu}%
473       \pgfpathmoveto{\pgfpoint\xister@TMPB\xister@Y}%
474       \pgfpathlineto{\pgfpoint\xister@TMPB\xister@XY@share}%
475       \pgfpathlineto{\pgfpoint\xister@X\xister@Y}%
476       \pgfusepath{stroke,clip}%
477       \pgfsetlinewidth{2\xister@Th}%
478       \xister@TMPA=\dimexpr\xister@Y\xister@sgn\xister@Th\relax
479       \pgfpathmoveto{\pgfpoint\z@\xister@TMPA}%
480       \pgfpathlineto{\pgfpoint\xister@X\xister@TMPA}%
481     \else\ifxister@five@dot@
482       \pgfpathcircle
483         {\pgfpoint{\dimexpr\xister@X-#3\xister@Five@dot\relax}%
484           {\dimexpr\xister@Y-#2\xister@Five@dot\relax}}%
485         {\xister@Five@dot}%
486       \pgfusepath{fill}%
487     \else
488       \xister@TMPB=\dimexpr\xister@X-#3\xister@Tv\relax
489       \pgfsetlinewidth{2\xister@Tv}%
490       \pgfpathmoveto{\pgfpoint\xister@TMPB\xister@Y}%
491       \pgfpathlineto
492         {%
493           \pgfpoint
494             \xister@TMPB
495             {\dimexpr\xister@Y-#2\xister@five@dash\xister@Tv*2\relax}%
496         }%
497     \fi\fi

```

```

498 }%
499 \xister@drawdigit{6#1}%
500 {%
501   \pgfsetlinewidth{2\xister@Tv}%
502   \xister@TMPB=\dimexpr\xister@X-#3\xister@Tv\relax
503   \pgfpathmoveto{\pgfpoint\xister@TMPB\xister@Y}%
504   \pgfpathlineto{\pgfpoint\xister@TMPB\xister@XY\share}%
505 }%
506 \xister@superimpose{7#1}%
507 {%
508   \rlap{\copy\csname xister@digitbox@1#1\endcsname}%
509   \copy\csname xister@digitbox@6#1\endcsname
510 }%
511 \xister@superimpose{8#1}%
512 {%
513   \rlap{\copy\csname xister@digitbox@2#1\endcsname}%
514   \copy\csname xister@digitbox@6#1\endcsname
515 }%
516 \ifxister@nine@triangle@
517 \xister@drawdigit{9#1}%
518 {%
519   \xister@TMPA=.5\dimexpr\xister@XY\share+\xister@Y\relax
520   \xister@TMPB=#3\dimexpr\xister@Ts-\xister@Tdd\relax
521   \xister@clip@v\z@
522   \pgfsetlinewidth{2\xister@Tdd}%
523   \pgfpathmoveto{\pgfpoint\xister@TMPB\xister@XY\share}%
524   \pgfpathlineto{\pgfpoint\xister@TMPB\xister@Y}%
525   \pgfpathlineto{\pgfpoint\xister@X\xister@TMPA}%
526   \pgfusepath{stroke}%
527   \xister@TMPB=#3\dimexpr\xister@Ts-\xister@Tdu\relax
528   \pgfsetlinewidth{2\xister@Tdu}%
529   \pgfpathmoveto{\pgfpoint\xister@TMPB\xister@Y}%
530   \pgfpathlineto{\pgfpoint\xister@TMPB\xister@XY\share}%
531   \pgfpathlineto{\pgfpoint\xister@X\xister@TMPA}%
532 }%
533 \else
534 \xister@superimpose{9#1}%
535 {%
536   \rlap{\copy\csname xister@digitbox@1#1\endcsname}%
537   \rlap{\copy\csname xister@digitbox@2#1\endcsname}%
538   \copy\csname xister@digitbox@6#1\endcsname
539 }%
540 \fi
541 \unless\ifxister@six@stroke@
542 \xister@drawdigit{6#1}%
543 {%
544   \ifxister@six@dot@
545     \xister@TMPA=\dimexpr\xister@X-#3\xister@Six@dot\relax
546     \pgfpathcircle
547       {\pgfpoint\xister@TMPA
548         {\dimexpr\xister@Y-#2\xister@Six@dot\relax}}%
549     {\xister@Six@dot}%
550     \pgfpathcircle
551     {\pgfpoint\xister@TMPA

```

```

552         {\dimexpr\xister@XY@share+#2\xister@Six@dot\relax}}%
553         {\xister@Six@dot}%
554         \pgfusepath{fill}%
555     \else
556         \xister@TMPB=\dimexpr\xister@X-#3\xister@Tv\relax
557         \xister@TMPA=2\dimexpr#2\xister@six@dash\xister@Tv\relax
558         \pgfsetlinewidth{2\xister@Tv}%
559         \pgfpathmoveto{\pgfpoint\xister@TMPB\xister@Y}%
560         \pgfpathlineto
561             {\pgfpoint\xister@TMPB
562              {\dimexpr\xister@Y-\xister@TMPA\relax}}%
563         \pgfpathmoveto{\pgfpoint\xister@TMPB\xister@XY@share}%
564         \pgfpathlineto
565             {\pgfpoint\xister@TMPB
566              {\dimexpr\xister@XY@share+\xister@TMPA\relax}}%
567     \fi
568 }%
569 \fi
570 }
571 \newcommand\xister@drawdigits@horizontal [3]
572 {%
573     \xister@XY@share=-\dimexpr\xister@share@mul\xister@X\xister@share@div\relax
574     \xister@drawdigit{1#1}%
575     {%
576         \pgfsetlinewidth{2\xister@Tv}%
577         \xister@TMPA=\dimexpr#3\xister@Tv-\xister@X\relax
578         \pgfpathmoveto{\pgfpoint\xister@TMPA\z@}%
579         \pgfpathlineto{\pgfpoint\xister@TMPA\xister@Y}%
580     }%
581     \xister@drawdigit{2#1}%
582     {%
583         \pgfsetlinewidth{2\xister@Tv}%
584         \xister@TMPA=\dimexpr\xister@XY@share-#3\xister@Tv\relax
585         \pgfpathmoveto{\pgfpoint\xister@TMPA\z@}%
586         \pgfpathlineto{\pgfpoint\xister@TMPA\xister@Y}%
587     }%
588     \xister@drawdigit{3#1}%
589     {%
590         \xister@TMPB=#2\dimexpr\xister@Ts-\xister@Tdu\relax
591         \xister@clip@h\xister@XY@share
592         \pgfsetlinewidth{2\xister@Tdu}%
593         \pgfpathmoveto{\pgfpoint\xister@XY@share\xister@TMPB}%
594         \pgfpathlineto{\pgfpoint{-\xister@X}\xister@TMPB}%
595         \pgfpathlineto{\pgfpoint\xister@XY@share\xister@Y}%
596     }%
597     \xister@drawdigit{4#1}%
598     {%
599         \xister@clip@h\z@
600         \xister@TMPB=#2\dimexpr\xister@Ts-\xister@Tdd\relax
601         \pgfsetlinewidth{2\xister@Tdd}%
602         \pgfpathmoveto{\pgfpoint\xister@X\xister@TMPB}%
603         \pgfpathlineto{\pgfpoint\xister@XY@share\xister@TMPB}%
604         \pgfpathlineto{\pgfpoint{-\xister@X}\xister@Y}%
605     }%

```

```

606 \xister@drawdigit{5#1}%
607   {%
608     \ifxister@five@triangle@
609       \xister@clip@h\z@
610       \xister@TMPB=#2\dimexpr\xister@Ts-\xister@Tdd\relax
611       \pgfsetlinewidth{2\xister@Tdd}%
612       \pgfpathmoveto{\pgfpoint{-\xister@X}\xister@TMPB}%
613       \pgfpathlineto{\pgfpoint\xister@XY@share\xister@TMPB}%
614       \pgfpathlineto{\pgfpoint{-\xister@X}\xister@Y}%
615       \pgfusepath{stroke,clip}%
616       \pgfsetlinewidth{2\xister@Tv}%
617       \xister@TMPA=\dimexpr#3\xister@Tv-\xister@X\relax
618       \pgfpathmoveto{\pgfpoint\xister@TMPA\z@}%
619       \pgfpathlineto{\pgfpoint\xister@TMPA\xister@Y}%
620     \else\ifxister@five@dot@
621       \pgfpathcircle
622         {\pgfpoint{\dimexpr#3\xister@Five@dot-\xister@X\relax}%
623           {\dimexpr\xister@Y-#2\xister@Five@dot\relax}}%
624         {\xister@Five@dot}%
625       \pgfusepath{fill}%
626     \else
627       \pgfsetlinewidth{2\xister@Th}%
628       \xister@TMPB=\dimexpr\xister@Y-#2\xister@Th\relax
629       \pgfpathmoveto{\pgfpoint{-\xister@X}\xister@TMPB}%
630       \pgfpathlineto
631         {%
632           \pgfpoint
633             {\dimexpr#3\xister@five@dash\xister@Th*2-\xister@X\relax}%
634             \xister@TMPB
635         }%
636       \fi\fi
637     }%
638 \xister@drawdigit{6#1}%
639   {%
640     \pgfsetlinewidth{2\xister@Th}%
641     \xister@TMPB=\dimexpr\xister@Y-#2\xister@Th\relax
642     \pgfpathmoveto{\pgfpoint{-\xister@X}\xister@TMPB}%
643     \pgfpathlineto{\pgfpoint\xister@XY@share\xister@TMPB}%
644   }%
645 \xister@superimpose{7#1}%
646   {%
647     \rlap{\copy\csname xister@digitbox@1#1\endcsname}%
648     \copy\csname xister@digitbox@6#1\endcsname
649   }%
650 \xister@superimpose{8#1}%
651   {%
652     \rlap{\copy\csname xister@digitbox@2#1\endcsname}%
653     \copy\csname xister@digitbox@6#1\endcsname
654   }%
655 \ifxister@nine@triangle@
656 \xister@drawdigit{9#1}%
657   {%
658     \xister@TMPA=.5\dimexpr\xister@XY@share-\xister@X\relax
659     \xister@TMPB=#2\dimexpr\xister@Ts-\xister@Tdu\relax

```



```

660         \xister@clip@h\z@
661         \pgfsetlinewidth{2\xister@Tdu}%
662         \pgfpathmoveto{\pgfpoint\xister@XY@share\xister@TMPB}%
663         \pgfpathlineto{\pgfpoint{-\xister@X}\xister@TMPB}%
664         \pgfpathlineto{\pgfpoint\xister@TMPA\xister@Y}%
665         \pgfusepath{stroke}%
666         \xister@TMPB=#2\dimexpr\xister@Ts-\xister@Tdd\relax
667         \pgfsetlinewidth{2\xister@Tdd}%
668         \pgfpathmoveto{\pgfpoint\xister@X\xister@TMPB}%
669         \pgfpathlineto{\pgfpoint\xister@XY@share\xister@TMPB}%
670         \pgfpathlineto{\pgfpoint\xister@TMPA\xister@Y}%
671     }%
672 \else
673     \xister@superimpose{9#1}%
674     {%
675         \rlap{\copy\csname xister@digitbox@1#1\endcsname}%
676         \rlap{\copy\csname xister@digitbox@2#1\endcsname}%
677         \copy\csname xister@digitbox@6#1\endcsname
678     }%
679 \fi
680 \unless\ifxister@six@stroke@
681     \xister@drawdigit{6#1}%
682     {%
683         \ifxister@six@dot@
684             \xister@TMPA=\dimexpr\xister@Y-#2\xister@Six@dot\relax
685             \pgfpathcircle
686                 {\pgfpoint{\dimexpr#3\xister@Six@dot-\xister@X\relax}%
687                  \xister@TMPA}%
688                 {\xister@Six@dot}%
689             \pgfpathcircle
690                 {\pgfpoint{\dimexpr\xister@XY@share-#3\xister@Six@dot\relax}%
691                  \xister@TMPA}%
692                 {\xister@Six@dot}%
693             \pgfusepath{fill}%
694         \else
695             \pgfsetlinewidth{2\xister@Th}%
696             \xister@TMPB=\dimexpr\xister@Y-#2\xister@Th\relax
697             \xister@TMPA=2\dimexpr#3\xister@six@dash\xister@Th\relax
698             \pgfpathmoveto{\pgfpoint{-\xister@X}\xister@TMPB}%
699             \pgfpathlineto
700                 {\pgfpoint{\dimexpr\xister@TMPA-\xister@X\relax}%
701                  \xister@TMPB}%
702             \pgfpathmoveto{\pgfpoint\xister@XY@share\xister@TMPB}%
703             \pgfpathlineto
704                 {\pgfpoint{\dimexpr\xister@XY@share-\xister@TMPA\relax}%
705                  \xister@TMPB}%
706         \fi
707     }%
708 \fi
709 }

```

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

```
710 \newcommand\xister@drawdigit[2]
```

```

711   {%
712   \expandafter\setbox\csname xister@digitbox@#1\endcsname=\hbox
713   {%
714   \begin{pgfpicture}%
715   \pgfpathrectanglecorners
716   {\pgfpoint{-\xister@bx\xister@X}{-\xister@by\xister@Y}}%
717   {\pgfpoint{\xister@bx\xister@X}{\xister@by\xister@Y}}%
718   \pgfusepath{use as bounding box}%
719   #2%
720   \pgfusepath{stroke}%
721   \pgfsetbaseline{\xister@sgn\xister@bs\xister@Y}%
722   \end{pgfpicture}%
723   }%
724   }

```

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

```

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

```

```

725 \newcommand*\xister@drawzero@vertical
726   {%
727   \xister@drawdigit{0}%
728   {%
729   \pgfsetlinewidth{2\xister@Ts}%
730   \pgfpathmoveto{\pgfpoint\z@{-\xister@Y}}%
731   \pgfpathlineto{\pgfpoint\z@\xister@Y}%
732   }%
733   }
734 \newcommand*\xister@drawzero@horizontal
735   {%
736   \xister@drawdigit{0}%
737   {%
738   \pgfsetlinewidth{2\xister@Ts}%
739   \pgfpathmoveto{\pgfpoint{-\xister@X}\z@}%
740   \pgfpathlineto{\pgfpoint\xister@X\z@}%
741   }%
742   }

```

(End of 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.

```

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

```

(End of 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 full expansion is used to evaluate the user specified sizes while the `\xister@font` is still active.

```

745 \protected\def\xister@setlengths
746   {%
747   \begingroup

```

```

748 \xister@font
749 \edef\xister@tmp
750   {%
751     \endgroup
752     \xister@X=\the\dimexpr.5\dimexpr\xister@x\relax\relax
753     \xister@Y=\the\dimexpr.5\dimexpr\xister@y\relax\relax
754     \xister@Th=\the\dimexpr.5\dimexpr\xister@th\relax\relax
755     \xister@Tv=\the\dimexpr.5\dimexpr\xister@tv\relax\relax
756     \xister@Tdu=\the\dimexpr.5\dimexpr\xister@tdu\relax\relax
757     \xister@Tdd=\the\dimexpr.5\dimexpr\xister@tdd\relax\relax
758     \xister@Five@dot=\the\dimexpr\xister@five@dot\relax
759     \xister@Six@dot=\the\dimexpr\xister@six@dot\relax
760     \ifx\xister@ts\@empty
761       \xister@Ts=%
762         \ifx\xister@build\xister@build@h\xister@Th\else\xister@Tv\fi
763     \else
764       \xister@Ts=\the\dimexpr.5\dimexpr\xister@ts\relax\relax
765     \fi
766   }%
767 \xister@tmp
768 }

```

(End of definition for \xister@setlengths.)

\xister@ensure@current The glyphs have to be rebuilt if the relevant font features changed (as specified in \xister@fontfeatures). If all these font features match with whatever was active when we last built the glyphs we do nothing, else we rebuild them.

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

```

769 \protected\edef\xister@ensure@current
770   {%
771     \unexpanded{\edef\xister@tmp{\xister@fontfeatures}}%
772     \unexpanded{\unless\ifx\xister@tmp\xister@last@font}%
773     \ifxister@dbg@
774       \unexpanded
775         {\xister@dbg@do{Font changed. Building glyphs for \xister@tmp}}%
776     \fi
777     \unexpanded
778     {%
779       \let\xister@last@font\xister@tmp
780       \xister@build
781     }%
782     \ifxister@dbg@
783       \unexpanded{\xister@dbg@done}%
784     \fi
785     \unexpanded{\fi}%
786   }
787 \AtBeginDocument{\xister@ensure@current}

```

(End of 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.

Symbols	E
\wedge 287	\edef 56, 58, 137, 157, 158, 159, 160, 161, 162, 163, 165, 167, 170, 175, 182, 199, 212, 219, 240, 247, 335, 749, 769, 771
A	\ekvcsvloop 204, 205
\AddToHook 123, 357, 363	\ekvdef 154, 208, 215, 236, 243, 324
\AtBeginDocument 787	\ekvdefNoVal 222, 223, 228, 250, 251, 256, 264, 265, 269, 275, 281, 314, 315, 318, 325
B	\ekvifdefined 324
\begin 714	\ekvifdefinedNoVal 325
\begingroup 112, 185, 286, 747	\ekvifdefinedset 150
C	\ekvlet 234, 235, 262, 263, 267, 268
\cistercian 333	\ekvletkv 155, 206, 207
\cistercianeval 343	\ekvletkvNoVal 316, 317
\cisterciannum 334, 335, 344	\ekvoProcessLocalOptions 313
\cisterciannumE 343	\ekvsetdef 233, 261, 266, 320
\cistercianredraw 314, 346	\else 117, 189, 374, 481, 487, 533, 555, 620, 626, 672, 694, 762, 763
\cistercianredrawlazy 315, 346	\end 722
\cisterciansetup 320, 331	\endcsname 41, 42, 43, 44, 47, 114, 118, 135, 144, 186, 190, 333, 339, 341, 405, 407, 409, 411, 416, 508, 509, 513, 514, 536, 537, 538, 647, 648, 652, 653, 675, 676, 677, 712, 744
\cistercianstyle 321	\endgroup 112, 185, 289, 751
\copy 405, 407, 409, 411, 412, 508, 509, 513, 514, 536, 537, 538, 647, 648, 652, 653, 675, 676, 677	\endinput 151
cs commands:	exp commands:
$\cs_new_protected:Npn$ 52	$\exp_args:No$ 127
\csname 41, 42, 43, 44, 47, 114, 118, 135, 144, 186, 190, 333, 339, 341, 405, 407, 409, 411, 416, 508, 509, 513, 514, 536, 537, 538, 647, 648, 652, 653, 675, 676, 677, 712, 744	\expandafter 41, 42, 43, 44, 47, 112, 113, 121, 135, 139, 140, 144, 185, 186, 190, 333, 334, 338, 341, 344, 373, 374, 379, 381, 416, 712, 744
D	\ExplSyntaxOff 46, 53, 132
\def 54, 61, 66, 71, 80, 84, 89, 94, 103, 110, 121, 133, 135, 152, 272, 273, 278, 279, 284, 291, 293, 294, 300, 306, 321, 327, 341, 342, 346, 348, 351, 352, 353, 354, 355, 360, 366, 369, 376, 385, 398, 399, 401, 745	\ExplSyntaxOn 38, 51, 122
\detokenize 114, 116, 118, 378	F
\dimexpr 297, 309, 420, 427, 433, 437, 444, 450, 460, 471, 478, 483, 484, 488, 495, 502, 519, 520, 527, 545, 548, 552, 556, 557, 562, 566, 573, 577, 584, 590, 600, 610, 617, 622, 623, 628, 633, 641, 658, 659, 666, 684, 686, 690, 696, 697, 700, 704, 752, 753, 754, 755, 756, 757, 758, 759, 764	\fi 119, 171, 191, 361, 367, 374, 382, 420, 427, 497, 540, 567, 569, 636, 679, 706, 708, 762, 765, 776, 784, 785
	H
	\hbox 712, 744
	I
	\if 378
	\ifdefined 359, 365

<code>\ifdim</code>	420, 427	<code>\pgfsetlinewidth</code>	436, 443,
<code>\ifx</code>	113, 171, 186, 373, 760, 762, 772		452, 461, 472, 477, 489, 501, 522,
int commands:			528, 558, 576, 583, 592, 601, 611,
<code>\int_step_inline:nm</code>	39		616, 627, 640, 661, 667, 695, 729, 738
L			
<code>\leavevmode</code>	371	<code>\pgfusepath</code>	422, 429, 476, 486,
<code>\let</code> ...	48, 49, 50, 63, 64, 82, 86, 87, 105,		526, 554, 615, 625, 665, 693, 718, 720
	107, 108, 109, 176, 177, 178, 179,	<code>\protected</code> ...	54, 61, 84, 110, 135, 154,
	190, 198, 204, 271, 277, 345, 356, 779		208, 215, 222, 223, 228, 233, 236,
<code>\long</code> ..	54, 110, 154, 208, 215, 236, 243,		243, 250, 251, 256, 261, 264, 265,
	284, 291, 293, 294, 300, 306, 324, 327		266, 269, 275, 281, 284, 291, 293,
			294, 300, 306, 314, 315, 320, 321,
			327, 331, 341, 346, 351, 401, 745, 769
M			
<code>\message</code>	288	<code>\ProvidesPackage</code>	1
N			
<code>\newcommand</code> 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,		R	
	14, 15, 16, 17, 18, 19, 333, 334, 343,	<code>\relax</code> .	115, 186, 297, 309, 344, 378, 379,
	417, 424, 431, 571, 710, 725, 734, 743		383, 398, 420, 427, 433, 437, 444,
<code>\newdimen</code>	26,		450, 460, 471, 478, 483, 484, 488,
	27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37		495, 502, 519, 520, 527, 545, 548,
<code>\newif</code>	20, 21, 22, 23, 24, 25		552, 556, 557, 562, 566, 573, 577,
<code>\newlinechar</code>	287		584, 590, 600, 610, 617, 622, 623,
<code>\newsavebox</code>	41, 42, 43, 44, 47		628, 633, 641, 658, 659, 666, 684,
<code>\normalfont</code>	13		686, 690, 696, 697, 700, 704, 752,
<code>\numexpr</code>	344, 379		753, 754, 755, 756, 757, 758, 759, 764
		<code>\RequirePackage</code>	2
		<code>\rlap</code>	354, 405, 407, 409, 411,
			508, 513, 536, 537, 647, 652, 675, 676
P			
<code>\PackageError</code>	116, 151, 187, 319, 330	S	
<code>\pgfpathcircle</code> 482, 546, 550, 621, 685, 689		<code>\setbox</code>	712, 744
<code>\pgfpathlineto</code>		str commands:	
	439, 446, 454, 455, 463, 464, 474,	<code>\str_if_in:nnTF</code>	127
	475, 480, 491, 504, 524, 525, 530,	T	
	531, 560, 564, 579, 586, 594, 595,	T _E X and L ^A T _E X 2 _ε commands:	
	603, 604, 613, 614, 619, 630, 643,	<code>\@backslashchar</code>	144
	663, 664, 669, 670, 699, 703, 731, 740	<code>\@cistercian</code>	333
<code>\pgfpathmoveto</code> ...	438, 445, 453, 462,	<code>\@empty</code>	171, 198, 204, 760
	473, 479, 490, 503, 523, 529, 559,	<code>\@firstofone</code>	383
	563, 578, 585, 593, 602, 612, 618,	<code>\@firstoftwo</code>	356
	629, 642, 662, 668, 698, 702, 730, 739	<code>\@gobble</code>	48, 49, 50, 64, 87
<code>\pgfpathrectanglecorners</code> ..	419, 426, 715	<code>\@gobbletwo</code>	381
<code>\pgfpoint</code>	420, 421, 427,	<code>\@ifpackageloaded</code>	125
	428, 438, 439, 445, 446, 453, 454,	<code>\@ifstar</code>	323
	455, 462, 463, 464, 473, 474, 475,	<code>\cisterciannum@pdf</code>	335
	479, 480, 483, 490, 493, 503, 504,	<code>\f@encoding</code>	145
	523, 524, 525, 529, 530, 531, 547,	<code>\f@family</code>	146
	551, 559, 561, 563, 565, 578, 579,	<code>\f@series</code>	147
	585, 586, 593, 594, 595, 602, 603,	<code>\f@shape</code>	148
	604, 612, 613, 614, 618, 619, 622,	<code>\f@size</code>	121, 149
	629, 632, 642, 643, 662, 663, 664,	<code>\ifxister@dbg@</code>	20, 773, 782
	668, 669, 670, 686, 690, 698, 700,	<code>\ifxister@five@dot@</code>	22, 481, 620
	702, 704, 716, 717, 730, 731, 739, 740	<code>\ifxister@five@triangle@</code>	21, 468, 608
<code>\pgfsetbaseline</code>	721	<code>\ifxister@nine@triangle@</code>	25, 516, 655
<code>\pgfsetcolor</code>	63, 64, 82, 86, 87, 105		

<code>\ifxister@six@dot@</code>	23, 544, 683	<code>\xister@font</code>	13, 182, 748
<code>\ifxister@six@stroke@</code>	24, 541, 680	<code>\xister@fontfeatures</code>	
<code>\xister@a</code>	341, 369	110, 137, 140, 204, 771
<code>\xister@add@fontfeature</code>	110, 204, 205	<code>\xister@fraction</code>	54, 196
<code>\xister@add@fontfeature@encoding</code>	133	<code>\xister@gobblecopy</code>	
<code>\xister@add@fontfeature@family</code>	133	352, 404, 406, 408, 410
<code>\xister@add@fontfeature@series</code>	133	<code>\xister@gobbletoat</code>	352, 387, 403
<code>\xister@add@fontfeature@shape</code>	133	<code>\xister@gobbletozero</code>	
<code>\xister@add@fontfeature@size</code>	133	352, 379, 404, 406, 408, 410
<code>\xister@b</code>	369	<code>\xister@if@slash</code>	51, 195
<code>\xister@bs</code>	12, 161, 721	<code>\xister@last@font</code>	348, 351, 772, 779
<code>\xister@build</code>	61, 190, 271, 277, 762, 780	<code>\xister@nine@triangle@false</code>	264
<code>\xister@build@h</code>	61, 762	<code>\xister@nine@triangle@true</code>	265
<code>\xister@build@horizontal</code>	61, 271	<code>\xister@pgfsetcolor</code>	63, 82, 86, 105
<code>\xister@build@v</code>	61	<code>\xister@setlengths</code>	65, 88, 745
<code>\xister@build@vertical</code>	61, 277	<code>\xister@sgn</code>	
<code>\xister@bx</code>	10, 159, 716, 717	66, 71, 80, 89, 94, 103, 437, 478, 721
<code>\xister@by</code>	11, 160, 716, 717	<code>\xister@share@div</code>	18, 58, 198, 433, 573
<code>\xister@c</code>	369	<code>\xister@share@mul</code>	19, 56, 199, 433, 573
<code>\xister@clip@h</code>	417, 591, 599, 609, 660	<code>\xister@six@dash</code>	17, 247, 557, 697
<code>\xister@clip@v</code>	417, 451, 459, 469, 521	<code>\xister@Six@dot</code>	37, 545, 548, 549,
<code>\xister@d</code>	369	552, 553, 684, 686, 688, 690, 692, 759
<code>\xister@dbg@dim</code>	48, 157, 158, 162,	<code>\xister@six@dot</code>	16, 240, 759
.	163, 165, 167, 171, 180, 213, 241, 294	<code>\xister@six@dot@false</code>	246, 259
<code>\xister@dbg@div</code>	49, 59, 306	<code>\xister@six@dot@true</code>	239, 254
<code>\xister@dbg@do</code>	291, 296, 302, 308, 775	<code>\xister@six@stroke@false</code>	
<code>\xister@dbg@done</code>	293, 298, 304, 310, 783	238, 245, 253, 258
<code>\xister@dbg@float</code>	48,	<code>\xister@six@stroke@true</code>	24, 250
.	57, 159, 160, 161, 200, 220, 248, 300	<code>\xister@stop</code>	54, 196
<code>\xister@dbg@msg</code>	284, 292, 293	<code>\xister@style</code>	321
<code>\xister@dbg@true</code>	283	<code>\xister@superimpose</code>	
<code>\xister@done</code>	352, 403	506, 511, 534, 645, 650, 673, 743
<code>\xister@drawdigit</code>	434, 441, 448, 457,	<code>\xister@symbols</code>	355, 380, 398, 399
.	466, 499, 517, 542, 574, 581, 588,	<code>\xister@Tdd</code>	
<code>\xister@drawdigits@horizontal</code>	597, 606, 638, 656, 681, 710, 727, 736	32, 73, 74, 96, 97, 450, 452, 520,
.	91, 95, 100, 104, 431	522, 600, 601, 610, 611, 666, 667, 757
<code>\xister@drawdigits@vertical</code>	68, 72, 77, 81, 431	<code>\xister@tdd</code>	6, 167, 178, 757
<code>\xister@drawzero@horizontal</code>	90, 725	<code>\xister@Tdu</code>	
<code>\xister@drawzero@vertical</code>	67, 725	31, 74, 75, 97, 98, 460, 461, 471,
<code>\xister@ensure@current</code>	349, 372, 769	472, 527, 528, 590, 592, 659, 661, 756
<code>\xister@five@dash</code>	15, 219, 495, 633	<code>\xister@tdu</code>	5, 165, 177, 756
<code>\xister@Five@dot</code>		<code>\xister@texorpdf</code>	337, 356
.	36, 483, 484, 485, 622, 623, 624, 758	<code>\xister@Th</code>	29, 436,
<code>\xister@five@dot</code>	14, 212, 758	437, 443, 444, 477, 478, 627, 628,
<code>\xister@five@dot@false</code>	218, 231	633, 640, 641, 695, 696, 697, 754, 762
<code>\xister@five@dot@true</code>	211, 226	<code>\xister@th</code>	3, 163, 176, 754
<code>\xister@five@triangle@false</code>		<code>\xister@tmp</code>	133, 144, 145,
.	210, 217, 225, 230	146, 147, 148, 149, 152, 157, 158,
<code>\xister@five@triangle@true</code>	21, 222	159, 160, 161, 162, 163, 164, 166,
		168, 173, 182, 183, 193, 203, 205,
		233, 234, 235, 261, 262, 263, 266,
		267, 268, 749, 767, 771, 772, 775, 779

<code>\xister@TMPA</code>	686, 698, 700, 716, 717, 739, 740, 752
.. 34, 73, 75, 96, 98, 297, 303, 309,	
437, 438, 439, 444, 445, 446, 478,	
479, 480, 519, 525, 531, 545, 547,	
551, 557, 562, 566, 577, 578, 579,	
584, 585, 586, 617, 618, 619, 658,	
664, 670, 684, 687, 691, 697, 700, 704	
<code>\xister@TMPB</code>	35, 450, 453,
454, 460, 462, 463, 470, 473, 474,	
488, 490, 494, 502, 503, 504, 520,	
523, 524, 527, 529, 530, 556, 559,	
561, 563, 565, 590, 593, 594, 600,	
602, 603, 610, 612, 613, 628, 629,	
634, 641, 642, 643, 659, 662, 663,	
666, 668, 669, 696, 698, 701, 702, 705	
<code>\xister@Ts</code>	33, 420,
427, 450, 460, 471, 520, 527, 590,	
600, 610, 659, 666, 729, 738, 761, 764	
<code>\xister@ts</code> ...	7, 170, 171, 179, 760, 764
<code>\xister@Tv</code>	30, 488,
489, 495, 501, 502, 556, 557, 558,	
576, 577, 583, 584, 616, 617, 755, 762	
<code>\xister@tv</code>	
.. 4, 162, 175, 176, 177, 178, 179, 755	
<code>\xister@X</code>	26,
69, 76, 78, 92, 99, 101, 420, 421, 428,	
439, 446, 455, 464, 475, 480, 483,	
488, 502, 525, 531, 545, 556, 573,	
577, 594, 602, 604, 612, 614, 617,	
622, 629, 633, 642, 658, 663, 668,	
	686, 698, 700, 716, 717, 739, 740, 752
<code>\xister@x</code>	8, 157, 272, 278, 752
<code>\xister@XY@share</code>	
.. 28, 433, 444, 451, 453, 455, 463,	
474, 504, 519, 523, 530, 552, 563,	
566, 573, 584, 591, 593, 595, 603,	
613, 643, 658, 662, 669, 690, 702, 704	
<code>\xister@Y</code>	27,
70, 79, 93, 102, 421, 427, 428, 433,	
437, 454, 462, 464, 473, 475, 478,	
484, 490, 495, 503, 519, 524, 529,	
548, 559, 562, 579, 586, 595, 604,	
614, 619, 623, 628, 641, 664, 670,	
684, 696, 716, 717, 721, 730, 731, 753	
<code>\xister@y</code>	9, 158, 273, 279, 753
<code>\z@</code>	303, 309, 420, 427, 438,
445, 459, 469, 479, 521, 578, 585,	
599, 609, 618, 660, 730, 731, 739, 740	
<code>\texorpdfstring</code>	359, 360, 365, 366
<code>\the</code>	334, 344, 379, 752,
753, 754, 755, 756, 757, 758, 759, 764	
tl commands:	
<code>\tl_if_in:nnTF</code>	52
	U
<code>\unexpanded</code>	56, 58, 121, 139,
140, 157, 158, 159, 160, 161, 162,	
163, 165, 167, 170, 175, 182, 199,	
212, 219, 240, 247, 296, 302, 308,	
337, 338, 771, 772, 774, 777, 783, 785	
<code>\unless</code>	171, 541, 680, 772