

The **GS1**^{*}package

GS1 Code Handler and Barcode Generator[†]

Markus Kohm[‡]

Released 2021/06/17

Abstract

There are several barcode packages out in the world, but they either need PStricks, or are restricted to EAN-13 barcodes. And most of all, they are all $\text{\LaTeX}\ 2\varepsilon$. I've decided to write a package, that supports several GS1 codes, and at almost the same time, I've decided to give L3 a chance. So I've started an experimental GS1 package using `expl3`. Using `expl3` was the main reason writing this package.

Contents

| | | |
|----------|--|----------|
| 1 | L3 Functions and Variables for GS1 Codes | 2 |
| 2 | $\text{\LaTeX}\ 2\varepsilon$ User Interface for GS1 Codes | 2 |
| 3 | Internal Functions and Variables | 4 |
| 4 | GS1 implementation | 5 |
| 4.1 | Implementation of Functions and Variables | 5 |
| 4.1.1 | Constants | 5 |
| 4.1.2 | Settings and Variables | 7 |
| 4.1.3 | Messages | 8 |
| 4.1.4 | Functions | 8 |
| 4.2 | Implementation of the User Interface | 14 |

Index

17

^{*}GS1 is a registered trademark of GS1 AISBL. Nevertheless, GS1 AISBL neither has any rights on this package, nor is responsible for it in any kind. The package's name should indicate only, that the package implements some aspects of GS1 codes.

[†]This file describes v23, last revised 2021/06/17.

[‡]E-mail: komascript@gmx.info

Preface

Design and implementation of this package based on

GS1: “Allgemeine GS1 Spezifikation”, Version 12.0, Januar 2012, Ausgabe 1.

This is the official GS1 specification for Germany, Austria and Switzerland.

Currently only EAN-8 and EAN-13 codes and bar codes without extension have been implemented. Others may follow in future.

1 L3 Functions and Variables for GS1 Codes

First of all: Please note, that the concept of private functions and variables is not well defined in TeX. Several variables, that I'd have made private in C++, haven't been declared to be private in this implementation. Maybe I should change this.

You should also know, that several test files may be created from the package source, and each of those may be used as an example for using the code. Nevertheless, LATEX 2 ϵ users will not need the following functions and should continue reading with [section 2](#).

`\GS_set_code_digit_seq:Nn \GS_set_code_digit_seq:Nn <GS1 sequence variable> {<token list>}`

Makes a GS1 sequence, that consist in digits only, from a *<token list>*. To do so, only the tokens from 0 up to 9 of the *<token list>* are set to the *<GS1 sequence variable>*. All other tokens are ignored. So may, e.g., convert the string “ISBN 978-3-86541-459-5” into a GS1 sequence with the digits “9783865414595”, where each digit is one item of the sequence.

`\GS_cut_EAN_control_digit:N \GS_cut_EAN_control_digit:N <GS1 sequence variable>`

The *<GS1 sequence variable>* should store either a EAN-8 or EAN-13 code with or without control digit. If the code has seven or twelve digits, nothing happens. If the code has eight or 13 digits, the last one will be removed. All other cases result in an error message.

`\GS_set_EAN_control_digit:N \GS_set_EAN_control_digit:N <GS1 sequence variable>`

The *<GS1 sequence variable>* should store either a EAN-8 or EAN-13 code with or without control digit. A new control digit will be calculated. If the code has seven or twelve digits the new control digit will be added. If the code has eight or 13 digits, the old control digit will be replaced by the new one. All other cases result in an error message.

`\int_set_to_EAN_control_digit:NN \int_set_to_EAN_control_digit:NN <integer variable> <GS1 sequence variable>`

Calculates the control digit of the *<GS1 sequence variable>* using the EAN control digit algorithm and stores it into the *<integer variable>*. Note, that the *<GS1 sequence variable>* may be a sequence of digits of any length not only seven digits for EAN-8 or twelve digits for EAN-13.

`\GS_use_as_EAN_barcode:N <GS1 sequence variable>`

Prints an EAN-8 or EAN-13 bar code depending on `\l_GS_code_size_int`. Note, that the *<GS1 sequence variable>* may have more than 8 resp. 13 items but not less! Use `\EANBarcode` if you need a more save function.

2 L^AT_EX 2 _{ϵ} User Interface for GS1 Codes

This section describes the L^AT_EX 2 _{ϵ} -compatible user interface. Note, that the test files `EANControlDigit.tex`, `EANBarcode.tex`, `GSSetup.tex`, and the resulting PDF files may be used as examples of the following commands.

`\EANControlDigit`

`\EANControlDigit{<string>}`

Only the digits of the `<string>` will be used. All other tokens will be ignored. If the `<string>` has 7 or 8 digits, the control digit of an EAN-8 code will be calculated and output. If the `<string>` has 12 or 13 digits, the control digit of an EAN-13 code will be calculated and output. If the `<string>` has 8 or 13 digits the last digit will be ignored. Any other number of digits will result in an error message.

`\EANBarcode`

`\EANBarcode[<options>]{<string>}`

Creates the EAN bar code corresponding with `<string>`. The optional argument `<options>` may be used to use different settings from the defaults set by `\GSSetup`.

Each digit of a EAN bar code is represented by seven modules. Each module is either black or white. A black module is a black, vertical line. A white module is just a gap. The seven modules start either with a black sequence of up to four modules, followed by a white sequence of up to four modules, followed by a black sequence of up to four modules, finished by a white sequence of up to four modules, or they start with a white sequence of up to four modules, followed by a black sequence of up to four modules, followed by a white sequence of up to four modules, finished by a black sequence of up to four modules.

| | |
|-----------------------|--|
| <code>\GSSetup</code> | <code>\GSSetup {<options>}</code> |
| | <p><code><options></code> is a list of <code><key>=<value></code> pairs. They are used to setup the default of several settings:</p> |
| | <code>ocrb=<boolean></code> |
| | <p>If <code><boolean></code> is <code>true</code> the digits at the bottom of the bar code will be printed using OCR-b font ocrb/T1/m/n in 9pt. Predefined default is <code>ocrb=true</code>.</p> |
| | <code>module_width=<dimension expression></code> |
| | <p>This is the width of one module. GS1 specifies a minimum module width of 0.264 mm and a normal width of 0.33 mm. You should not set a width below the minimum!</p> |
| | <code>module_height=<dimension expression></code> |
| | <p>This is the height of a black module. GS1 specifies a normal bar code height of 21.31 mm for EAN-8 and 25.01 mm for EAN-13. Both values are inclusive the digits at the bottom of the bar code. Some marker modules are higher than the digit modules.</p> |
| | <code>code=<string></code> |
| | <p>The <code><string></code> should either be EAN-8 or EAN-13. The predefined default is EAN-13. More types will be supported in future.</p> |
| | <code>scale=<floating point></code> |
| | <p>This is the scale factor for the bar code. GS1 specifies scale classes from 0.8 up to 2.0 with steps of 0.05. Factors less than 0.8 shouldn't be used. <i>Currently scale won't be used!</i></p> |
| | <code>scale_to_font=<boolean></code> |
| | <p>Ignore <code>module_width</code> and instead set the module width depending on the width of digit 0 of the current font. Note, that this will not scale the whole bar code but only the module width. To scale the whole bar code, you should use <code>scale</code>.</p> |
| | <code>add_control=<boolean></code> |
| | <p>Add the control digit to the GS1 code. If there's already a control digit, replace it by the calculated one. The predefined default is <code>add_control=false</code>.</p> |

3 Internal Functions and Variables

You should not use or manipulate these! So, maybe it's better to stop reading now.

| | |
|------------------------------------|---|
| <code>__GS_set_key_code:nn</code> | <code>__GS_set_key_code:nn {<token list>} {<integer expression>}</code> |
| | <p>Sets <code>\l_GS_code_type_t1</code> to <code><token list></code> and <code>\l_GS_code_size_int</code> to value of <code>{<integer expression>}</code>.</p> |
| <code>__GS_new_seq_c:cn</code> | <code>__GS_new_seq_c:cn {<sequence name>} {<token list>}</code> |
| | <p>Creates a sequence constant <code>\c__GS_<sequence name>_seq</code>. The value of the constant will be build by the tokens of the <code><token list></code>. These tokens should be either characters "A" or "B" for selection constants or digits 1–4 for module constants.</p> |

```
\_\_GS\_modules:Nn          \_\_GS\_modules:Nn <sequence variable> {\<dimension expression>}
\_\_GS\_modules:cn          \_\_GS\_modules:cn {\<sequence variable name>} {\<dimension expression>}
\_\_GS\_modules:NnN         \_\_GS\_modules:NnN <sequence variable> {\<dimension expression>} <boolean variable>

```

Draws the modules given by the *<sequence variable>* with height *{<dimension expression>}*.
The arguments are:

- #1 : *<sequence variable>* or *<sequence variable name>*, each item of the sequence stays for a number of modules with the same color. $\backslash 1_GS_black_bool$ signals, whether the (first) modules are black or white and will be reversed after every item. Each module has the width $\backslash 1_GS_module_wd_dim$.
- #2 : *<dimension expression>*, the height of the black modules. The modules will be raised by $\backslash 1_GS_module_ht_dim$.
- #3 : *<boolean variable>*, true indicates, that the first module should be black. With false, the first module will be white.

```
\_\_GS\_modules_start_black:Nn \_\_GS\_modules_start_black:Nn <sequence variable> {\<dimension expression>}

```

Same like $\backslash 1_GS_modules:NnN <sequence variable> {\<dimension expression>} \c_true_bool$.

```
\_\_GS\_modules_start_white:Nn \_\_GS\_modules_start_white:Nn <sequence variable> {\<dimension expression>}

```

Same like $\backslash 1_GS_modules:NnN <sequence variable> {\<dimension expression>} \c_false_bool$.

4 GS1 implementation

1 $\langle @@=GS \rangle$

The implementation has been done in two parts. The first part is the L3 code with all the functions and variables. The second part is the LATEX 2 ε lookalike user interface.
But before this, we just declare, what this is:

2 $\langle *package \rangle$
 3 $\backslash ProvidesExplPackage$
 4 $\{ \backslash ExplFileName \} \{ \backslash ExplFileDate \} \{ \backslash ExplFileVersion \} \{ \backslash ExplFileDescription \}$

and what it requires:

5 $\backslash RequirePackage \{ rule-D \}$
 6 $\langle /package \rangle$

4.1 Implementation of Functions and Variables

4.1.1 Constants

$\backslash 1_GS_new_seq_c:cn$ While this is an internal function, that should allow only some tokens at the arguments, it is declared **nopar**.

7 $\langle *package \rangle$
 8 $\backslash cs_new_nopar:Npn \backslash 1_GS_new_seq_c:cn \#1 \#2$
 9 $\{$
 10 $\quad \backslash seq_new:c \{ c_GS_ \#1 _seq \}$
 11 $\quad \backslash seq_set_split:Nnn \backslash l_tmpa_seq \{ \} \{ \#2 \}$
 12 $\quad \backslash seq_gset_eq:cN \{ c_GS_ \#1 _seq \} \backslash l_tmpa_seq$
 13 $\}$
 14 $\langle /package \rangle$

(End definition for `_GS_new_seq_c:cn`.)

```
\c_GS_AB0_seq
\c_GS_AB1_seq
\c_GS_AB2_seq
15 <*package>
\c_GS_AB3_seq
16 \_GS_new_seq_c:cn {AB0} {AAAAAA}
\c_GS_AB4_seq
17 \_GS_new_seq_c:cn {AB1} {AABABB}
\c_GS_AB5_seq
18 \_GS_new_seq_c:cn {AB2} {AABBAB}
\c_GS_AB6_seq
19 \_GS_new_seq_c:cn {AB3} {AABBBA}
\c_GS_AB7_seq
20 \_GS_new_seq_c:cn {AB4} {ABAABB}
\c_GS_AB8_seq
21 \_GS_new_seq_c:cn {AB5} {ABBAAB}
\c_GS_AB9_seq
22 \_GS_new_seq_c:cn {AB6} {ABBBAA}
23 \_GS_new_seq_c:cn {AB7} {ABABAB}
24 \_GS_new_seq_c:cn {AB8} {ABABBA}
25 \_GS_new_seq_c:cn {AB9} {ABBABA}
26 </package>
```

(End definition for `\c_GS_AB0_seq` and others.)

`\c_GS_A0_seq`
`\c_GS_A1_seq`
`\c_GS_A2_seq`
`\c_GS_A3_seq`
`\c_GS_A4_seq`
`\c_GS_A5_seq`
`\c_GS_A6_seq`
`\c_GS_A7_seq`
`\c_GS_A8_seq`
`\c_GS_A9_seq`
`\c_GS_B0_seq`
`\c_GS_B1_seq`
`\c_GS_B2_seq`
`\c_GS_B3_seq`
`\c_GS_B4_seq`
`\c_GS_B5_seq`
`\c_GS_B6_seq`
`\c_GS_B7_seq`
`\c_GS_B8_seq`
`\c_GS_B9_seq`

```
\c_GS_margin_seq
\c_GS_separator_seq
\c_GS_special_seq
\c_GS_extra_margin_seq
\c_GS_extra_separator_seq
48 \_GS_new_seq_c:cn {margin} {111} % start black
49 \_GS_new_seq_c:cn {separator} {11111} % start white
50 \_GS_new_seq_c:cn {special} {111111} % start white
51 \_GS_new_seq_c:cn {extra_margin} {112} % start black
52 \_GS_new_seq_c:cn {extra_separator} {11} % start white
53 </package>
```

(End definition for `\c_GS_A0_seq` and others.)

There are some basic dimensions for the modules at the specification:

```

\c__GS_module_min_width_dim
\c__GS_module_norm_width_dim
54  {*package}
55  \dim_const:Nn \c__GS_module_min_width_dim {0.264mm}
56  \dim_const:Nn \c__GS_module_norm_width_dim {0.33mm}
57  
```

(End definition for `\c__GS_module_min_width_dim` and `\c__GS_module_norm_width_dim`.)

4.1.2 Settings and Variables

These settings will influence the work of several of the user functions. They are defined as keys of family GS1.

`\l_GS_use_ocrb_bool` Needed to set both `\l_GS_code_type_tl` and `\l_GS_code_size_int` with one key. Together they are the type of code, to be handled.

```

\l_GS_module_wd_dim
\l_GS_module_ht_dim
\l_GS_scale_fp
\l_GS_scale_to_font_bool
\l_GS_add_control_bool
\l_GS_code_type_tl
\l_GS_code_size_int
\_\_GS_set_key_code:nn
58  {*package}
59  \cs_new_nopar:Npn \_\_GS_set_key_code:nn #1#2
60  {
61      \tl_if_exist:NF \l_GS_code_type_tl { \tl_new:N \l_GS_code_type_tl }
62      \tl_set:Nn \l_GS_code_type_tl { #1 }
63      \int_if_exist:NF \l_GS_code_size_int { \int_new:N \l_GS_code_size_int }
64      \int_set:Nn \l_GS_code_size_int { #2 }
65  }
66  \keys_define:nn { GS1 }
67  {
68      ocrb          .bool_set:N = \l_GS_use_ocrb_bool,
69      ocrb          .initial:n = true,
70      module_width .dim_set:N = \l_GS_module_wd_dim,
71      module_width .initial:V = \c__GS_module_norm_width_dim,
72      module_height.dim_set:N = \l_GS_module_ht_dim,
73      module_height.initial:V = \c_zero_dim,
74      code           .choice:,
75      code / EAN-8  .code:n = { \_\_GS_set_key_code:nn { EAN } { 8 } },
76      code / EAN8   .code:n = { \_\_GS_set_key_code:nn { EAN } { 8 } },
77      code / EAN-13 .code:n = { \_\_GS_set_key_code:nn { EAN } { 13 } },
78      code / EAN13  .code:n = { \_\_GS_set_key_code:nn { EAN } { 13 } },
79      code           .initial:n = EAN-13,
80      scale          .fp_set:N = \l_GS_scale_fp,
81      scale          .initial:n = 1.0,
82      scale_to_font .bool_set:N = \l_GS_scale_to_font_bool,
83      scale_to_font .initial:n = false,
84      add_control   .bool_set:N = \l_GS_add_control_bool,
85      add_control   .initial:n = false,
86  }
87 
```

(End definition for `\l_GS_use_ocrb_bool` and others. These variables are documented on page ??.)

Note: Later I'll define a L^AT_EX 2 _{ε} command to change the defaults of those keys. Additionally local changes of those keys may be done using the optional argument of the L^AT_EX 2 _{ε} user commands. See [subsection 4.2](#) for more information.

`\l__GS_code_seq` This will be used later for several local GS1 sequences. It is private and also shouldn't be used in global context.

```

88  (*package)
89  \seq_new:N \l__GS_code_seq
90  (/package)

```

(End definition for `\l__GS_code_seq`.)

4.1.3 Messages

We need a message for not supported lengths of EAN codes, because currently only EAN-8 and EAN-13 are supported, both with or without control digit. This message will be used as an error message.

```

91  (*package)
92  \msg_new:nnn { GS1 } { EAN-code-size }
93  {
94      #1~isn't~a~valid~EAN~code~\msg_line_context:..
95  }
96  {
97      The~given~code~is~neither~a~EAN-8~with~or~without~control~digit,\\
98      nor~a~EAN-13~with~or~without~control~digit.\\\\\
99      The~GS1~module~currently~only~supports~EAN-8~and~EAN-13.
100 }
101 (/package)

```

Another message is only a warning message. It will be used whenever the used module width would be less than the minimum module width given by the GS1 specification.

```

102 (*package)
103 \msg_new:nnn { GS1 } { module/minwidth }
104 {
105     Resulting~module~width~is~less~than~allowed~minimum~\msg_line_context:.\\\\\
106     GS1~specification~declares~a~minimum~module~width~of~#2.\\\
107     Currently~the~module~with~would~be~#1.\\\
108     To~avoid~problems,~I'll~increase~module~width~to~#2.
109 }
110 (/package)

```

4.1.4 Functions

`\GS_set_code_digit_seq:Nn` Convert a string into a code sequence ignoring all but digits.

```

111 (*package)
112 \cs_new_nopar:Npn \GS_set_code_digit_seq:Nn #1#2
113 {
114     \seq_set_eq:NN #1 \c_empty_seq
115     \tl_map_inline:nn
116         { #2 }
117     {
118         \tl_if_in:nnT
119             {0123456789}
120             {##1}
121         {
122             \seq_put_right:Nn #1 { ##1 }
123         }
124     }
125 }
126 (/package)

```

The following test files are used for this code: *GS_set_code_digit.tex*.

```

127  {*test & GS.set.code.digit.seq}
128  \ExplSyntaxOn
129  \seq_new:N \l_testa_seq
130  \seq_new:N \l_testb_seq
131
132  \GS_set_code_digit_seq:Nn \l_testa_seq {ISBN 978-3-86541-459-5}
133  Sequence~for~ISBN~978-3-86541-459-5~is:~
134  \seq_use:Nnnn \l_testa_seq { ,~ } { ,~ } { ,~ }
135
136  \GS_set_code_digit_seq:Nn \l_testb_seq {9783865414595}
137  \seq_map_inline:Nn \l_testa_seq
138  {
139      \seq_pop_left:NN \l_testb_seq \l_tmpa_tl
140      \tl_set:Nn \l_tmpb_tl { #1 }
141      \tl_if_eq:NNF \l_tmpa_tl \l_tmpb_tl
142      {
143          \tl_show:N \l_testa_seq
144          \tl_show:N \l_testb_seq
145          \msg_fatal:nnn { GS1/test } { function } { \GS_set_code_digit_seq:Nn }
146      }
147  }
148
149  \GS_set_code_digit_seq:Nn \l_testa_seq {ISBN 978-3-86541-459-5}
150  \seq_set_split:Nnn \l_testb_seq {,} {9,7,8,3,8,6,5,4,1,4,5,9,5}
151  \seq_map_inline:Nn \l_testa_seq
152  {
153      \seq_pop_left:NN \l_testb_seq \l_tmpa_tl
154      \tl_set:Nn \l_tmpb_tl { #1 }
155      \tl_if_eq:NNF \l_tmpa_tl \l_tmpb_tl
156      {
157          \tl_show:N \l_testa_seq
158          \tl_show:N \l_testb_seq
159          \msg_fatal:nnn { GS1/test } { function } { \GS_set_code_digit_seq:Nn }
160      }
161  }
162
163  \ExplSyntaxOff
164  /test & GS.set.code.digit.seq

```

(End definition for *\GS_set_code_digit_seq:Nn*. This function is documented on page 2.)

\GS_cut_EAN_control_digit:N EAN code sequences with control digit are either 8 or 13 digits. To remove the control digit we just have to remove the right most digit from a 8 or 13 digits sequence. 7 or 12 digit sequences are already without control digit. All other sequences are not supported.

```

165  (*package)
166  \cs_new_nopar:Npn \GS_cut_EAN_control_digit:N #1
167  {
168      \int_case:nnF
169      { \seq_count:N #1 }
170      {
171          { 7 } { }
172          { 8 } { \seq_pop_right:NN #1 \l_tempa_tl }
173          { 12 } { }

```

```

174     { 13 } { \seq_pop_right:NN #1 \l_tempa_tl }
175   }
176   {
177     \msg_error:nnn { GS1 } { EAN-code-size } { #1 }
178   }
179 }
180 
```

The following test files are used for this code: *GS_cut_EAN_control_digit.tex*.

```

181 (*test & GS.cut.EAN.control.digit)
182 \raggedright
183 \ExplSyntaxOn
184 \seq_new:N \l_testa_seq
185 \GS_set_code_digit_seq:Nn \l_testa_seq {ISBN 978-3-86541-459-5}
186 With~control:~\seq_use:Nnnn \l_testa_seq { ,~ } { ,~ } { ,~ }\\
187 \GS_set_code_digit_seq:Nn \l_testb_seq {ISBN 978-3-86541-459}
188 \GS_cut_EAN_control_digit:N \l_testa_seq
189 \seq_map_inline:Nn \l_testa_seq
190   {
191     \seq_pop_left:NN \l_testb_seq \l_tmpa_tl
192     \tl_set:Nn \l_tmpb_tl { #1 }
193     \tl_if_eq:NNF \l_tmpa_tl \l_tmpb_tl
194     {
195       \tl_show:N \l_testa_seq
196       \tl_show:N \l_testb_seq
197       \msg_fatal:nnn { GS1/test } { function }
198       { \GS_cut_EAN_control_digit:N }
199     }
200   }
201 Without~control:~\seq_use:Nnnn \l_testa_seq { ,~ } { ,~ } { ,~ }\\
202 \ExplSyntaxOff
203 
```

(End definition for *\GS_cut_EAN_control_digit:N*. This function is documented on page 2.)

\int_set_to_EAN_control_digit:NN Sets an integer to the control digit calculated with the EAN control digit algorithm for a given code sequence. Note, that the complete code sequence will be used to calculate the control digit. So, if you have a EAN-8 or EAN-13 code sequence, you should cut off the control digit first.

```

204 (*package)
205 \cs_new_nopar:Npn \int_set_to_EAN_control_digit:NN #1#2
206   {
207     \int_zero:N #1
208     \seq_set_eq:NN \l_tmpa_seq #2
209     \bool_until_do:nn
210     { \seq_if_empty_p:N \l_tmpa_seq }
211     {
212       \seq_pop_left:NN \l_tmpa_seq \l_tmpb_tl
213       \int_if_even:nTF
214       { \seq_count:N \l_tmpa_seq }
215       {
216         \int_add:Nn #1 { 3 * \l_tmpb_tl }
217       }
218     }

```

```

219           \int_add:Nn #1 { \l_tmpb_tl }
220       }
221   }
222   \int_set:Nn #1 { \int_mod:nn { 10 - \int_mod:nn { #1 } { 10 } } { 10 } }
223 }
224 
```

The following test files are used for this code: *int_set_to_EAN_control_digit.tex*.

```

225 {*test & int.set.to.EAN.control.digit}
226 \ExplSyntaxOn
227 \seq_new:N \l_testa_seq
228 \tl_new:N \l_control_tl
229 \GS_set_code_digit_seq:Nn \l_testa_seq {ISBN 978-3-86541-459-5}
230 \seq_pop_right:NN \l_testa_seq \l_control_tl
231 \int_new:N \l_control_int
232 \int_set_to_EAN_control_digit:NN \l_control_int \l_testa_seq
233 Control-digit~should~be~\tl_use:N \l_control_tl{} ~
234 \int_compare:nNnTF { \l_control_tl } { = } { \l_control_int }
235 {
236     ~and~is~\int_use:N \l_control_int .
237 }
238 {
239     ~but~it~\int_use:N \l_control_int .
240     \msg_error:nnn { GS1/test } { function }
241     { \int_set_to_EAN_control_digit:NN }
242 }
243 \ExplSyntaxOff
244 
```

(End definition for *\int_set_to_EAN_control_digit:NN*. This function is documented on page 2.)

\GS_set_EAN_control_digit:N Add a new control digit to a EAN sequence

```

245 {*package}
246 \cs_new_nopar:Npn \GS_set_EAN_control_digit:N #1
247 {
248     \GS_cut_EAN_control_digit:N #1
249     \int_set_to_EAN_control_digit:NN \l_tmpa_int #1
250     \seq_put_right:NV #1 \l_tmpa_int
251 }
252 
```

The following test files are used for this code: *GS_set_EAN_control_digit.tex*.

```

253 {*test & GS.set.EAN.control.digit}
254 \ExplSyntaxOn
255 \seq_new:N \l_testa_seq
256 \GS_set_code_digit_seq:Nn \l_testa_seq {ISBN 978-3-86541-459-5}
257 \seq_new:N \l_testb_tl
258 \seq_pop_right:NN \l_testa_seq \l_testb_tl
259 \GS_set_EAN_control_digit:N \l_testa_seq
260 \tl_new:N \l_testa_tl
261 \seq_get_right:NN \l_testa_seq \l_testa_tl
262 Control-digit~should~be~\tl_use:N \l_testb_tl {} ~
263 \int_compare:nNnTF { \l_testa_tl } { = } { \l_testb_tl }
264 {
265     ~and~is~\tl_use:N \l_testa_tl .

```

```

266   }
267   {
268     ~but~it~\tl_use:N \t_testa_tl .
269     \msg_error:nnn { GS1/test } { function } { \GS_set_EAN_control_digit:N }
270   }
271 \ExplSyntaxOff
272 
```

(End definition for `\GS_set_EAN_control_digit:N`. This function is documented on page 2.)

```

\__GS_modules:Nn
\__GS_modules:cn
\__GS_modules:NnN
\__GS_modules_start_black:Nn
\__GS_modules_start_white:Nn
273 (*package)
274 \cs_new_nopar:Npn \__GS_modules:Nn #1#2
275   {
276     \seq_map_inline:Nn #1
277     {
278       \bool_if:NTF \l__GS_black_bool
279       {
280         \bool_set_false:N \l__GS_black_bool
281         \hbox_set:Nn \l_tmpa_box { 0 }
282         \rule:nnn
283         {
284           \dim_eval:n
285           {
286             \box_ht:N \l_tmpa_box + \l_GS_module_wd_dim / 2
287             - \dim_eval:n { #2 } + \l_GS_module_ht_dim
288           }
289         }
290         { \dim_eval:n { \l_GS_module_wd_dim * ##1 } }
291         { \dim_eval:n { #2 } }
292       }
293     }
294     \bool_set_true:N \l__GS_black_bool
295     \hbox_to_wd:nn { \l_GS_module_wd_dim * ##1 } { }
296   }
297 }
298 }
299
300 \cs_new_nopar:Npn \__GS_modules:NnN #1#2#3
301   {
302     \bool_if_exist:NF \l__GS_black_bool { \bool_new:N \l__GS_black_bool }
303     \bool_set_eq:NN \l__GS_black_bool #3
304     \__GS_modules:Nn #1 { #2 }
305   }
306
307 \cs_new_nopar:Npn \__GS_modules_start_black:Nn #1#2
308   {
309     \__GS_modules:NnN #1 { #2 } \c_true_bool
310   }
311
312 \cs_new_nopar:Npn \__GS_modules_start_white:Nn #1#2
313   {
314     \__GS_modules:NnN #1 { #2 } \c_false_bool
315   }

```

```

316
317 \cs_generate_variant:Nn \__GS_modules:Nn { c }
318 
```

The following test files are used for this code: EANBarcode.tex.

(End definition for __GS_modules:Nn and others.)

\GS_use_as_EAN_barcode:N Puts the digits, rules, and gaps for an EAN barcode into the input stream.

```

319 {*package}
320 \cs_new_nopar:Npn \GS_use_as_EAN_barcode:N #1
321 {
322   \seq_set_eq:NN \l__GS_code_seq #1
323
324   \int_compare:nNnTF { \l_GS_code_size_int } { = } { 8 }
325   {
326     \hbox_to_wd:nn { \l_GS_module_wd_dim * 7 } { }
327     \seq_set_eq:Nc \l_GS_system_seq { c__GS_AB0_seq }
328   }
329   {
330     \hbox_to_wd:nn { \l_GS_module_wd_dim * 11 } { }
331     \seq_pop_left:NN \l_GS_code_seq \l_tmpa_tl
332     \seq_set_eq:Nc \l_GS_system_seq { c__GS_AB \l_tmpa_tl _seq }
333     \hbox_overlap_left:n { \l_tmpa_tl }
334   }
335
336   \__GS_modules_start_black:Nn \c__GS_margin_seq
337   { \l_GS_module_ht_dim + \l_GS_module_wd_dim * 5 }
338
339   \int_step_inline:nnnn { 1 } { 1 }
340   { \int_div_truncate:nn { \l_GS_code_size_int } { 2 } }
341   {
342     \seq_pop_left:NN \l_GS_code_seq \l_tmpa_tl
343     \hbox_overlap_right:n { \l_tmpa_tl }
344     \seq_pop_left:NN \l_GS_system_seq \l_tmpb_tl
345     \__GS_modules:cn { c__GS_ \l_tmpb_tl \l_tmpa_tl _seq }
346     { \l_GS_module_ht_dim }
347   }
348
349   \__GS_modules_start_white:Nn \c__GS_separator_seq
350   { \l_GS_module_ht_dim + \l_GS_module_wd_dim * 5 }
351
352   \int_step_inline:nnnn { 1 } { 1 }
353   { \int_div_truncate:nn { \l_GS_code_size_int } { 2 } }
354   {
355     \seq_pop_left:NN \l_GS_code_seq \l_tmpa_tl
356     \hbox_overlap_right:n { \l_tmpa_tl }
357     \__GS_modules:cn { c__GS_A \l_tmpa_tl _seq }
358     { \l_GS_module_ht_dim }
359   }
360
361   \__GS_modules_start_black:Nn \c__GS_margin_seq
362   { \l_GS_module_ht_dim + \l_GS_module_wd_dim * 5 }
363
364   \hbox_to_wd:nn { \l_GS_module_wd_dim * 7 } { }

```

```

365   }
366 
```

The following test files are used for this code: EANBarcode.tex.

(End definition for \GS_use_as_EAN_barcode:N. This function is documented on page 2.)

4.2 Implementation of the User Interface

For this, additional packages are needed:

```

367 {*package}
368 \RequirePackage{xparse}
```

\EANControlDigit

```

369 \NewDocumentCommand \EANControlDigit
370   { m }
371   {
372     \group_begin:
373       \GS_set_code_digit_seq:Nn \l__GS_code_seq { #1 }
374       \GS_cut_EAN_control_digit:N \l__GS_code_seq
375       \int_set_to_EAN_control_digit:NN \l_tmpa_int \l__GS_code_seq
376       \int_to_arabic:n { \l_tmpa_int }
377     \group_end:
378   }
379 
```

The following test files are used for this code: EANControlDigit.tex.

```

380 {*test & EANControlDigit}
381 \begin{tabular}{ll}
382   \hline
383   Calculated & Known \\
384   \hline
385   501234567890-\EANControlDigit{501234567890} & 501234567890-0 \\
386   ISBN 978-3-86541-459-\EANControlDigit{ISBN 978-3-86541-459} & ISBN
387   978-3-86541-459-5 \\
388   EAN-8: 2012345\EANControlDigit{2012345} & EAN-8: 20123451 \\
389   \hline
390 \end{tabular}
391 
```

(End definition for \EANControlDigit. This function is documented on page 3.)

\EANBarcode

```

392 {*package}
393 \NewDocumentCommand \EANBarcode
394   { o m }
395   {
396     \group_begin:
397       \IfNoValueF{#1}{\keys_set:nn { GS1 } { #1 } }
398       \dim_compare:nNnT { \l__GS_module_ht_dim } { = } { \c_zero_dim }
399       {
400         \int_compare:nNnTF { \l__GS_code_size_int } { = } { 8 }
401         { \dim_set:Nn \l__GS_module_ht_dim { 21.31 mm } }
402         { \dim_set:Nn \l__GS_module_ht_dim { 25.01 mm } }
403       }

```

```

404     \bool_if:nT \l_GS_use_ocrb_bool
405     {
406         \usefont{OT1}{ocrb}{m}{n}\fontsize{9}{9}\selectfont
407     }
408
409     \GS_set_code_digit_seq:Nn \l__GS_code_seq { #2 }
410
411     \bool_if:NT \l_GS_add_control_bool
412     {
413         \GS_set_EAN_control_digit:N \l__GS_code_seq
414     }
415
416     \int_compare:nNnT
417     {
418         \seq_count:N \l__GS_code_seq
419     }
420     {
421         \msg_error:nnn { GS1 } { EAN-code-size } { #2 }
422     }
423     \int_while_do:nNnn
424     {
425         \seq_count:N \l__GS_code_seq
426     }
427     {
428         \seq_put_left:Nn \l__GS_code_seq { 0 }
429     }
430
431     \bool_if:NT \l_GS_scale_to_font_bool {
432         \hbox_set:Nn \l_tmpa_box { 0 }
433         \dim_set:Nn \l_GS_module_wd_dim { \box_wd:N \l_tmpa_box / 7 }
434     }
435
436     \dim_set:Nn \l_tmpa_dim
437     {
438         \fp_to_decimal:N \l_GS_scale_fp \l_GS_module_wd_dim
439     }
440     \dim_compare:nNnT
441     {
442         \l_tmpa_dim
443     }
444     {
445         \msg_warning:nnxx { GS1 } { module/minwidth }
446         {
447             \dim_use:N \l_GS_module_wd_dim
448             \dim_set:Nn \l_GS_module_wd_dim
449             {
450                 \c__GS_module_min_width_dim *
451                 100 / \fp_to_int:n { 100 * \l_GS_scale_fp }
452             }
453         }
454     \hbox_set:Nn \l_tmpa_box { \GS_use_as_EAN_barcode:N \l__GS_code_seq }
455     \box_scale:Nnn \l_tmpa_box
456     {
457         \fp_to_int:n { 100 * \l_GS_scale_fp } / 100

```

```

457     { \fp_to_int:n { 100 * \l_gs_scale_fp } / 100 }
458     \box_use:N \l_tmpa_box
459 \group_end:
460 }
461 
```

The following test files are used for this code: *EANBarcode.tex*.

```

462 {*test & EANBarcode | GSSetup}
463 \raggedright
464 \verb|\EANBarcode{ISBN 978-3-86541-459-5}|:
465     \EANBarcode{ISBN 978-3-86541-459-5}
466
467 \verb|\EANBarcode[add_control]{ISBN 978-3-86541-459-5}|:
468     \EANBarcode[add_control]{ISBN 978-3-86541-459-5}
469
470 \verb|\EANBarcode[ocrb=false]{ISBN 978-3-86541-459-5}|:
471     \EANBarcode[ocrb=false]{ISBN 978-3-86541-459-5}
472
473 \verb|\EANBarcode[code=EAN-8]{20123451}|:
474     \EANBarcode[code=EAN-8]{20123451}
475
476 \verb|\EANBarcode[code=EAN-8,ocrb=false,add_control]{2012345}|:
477     \EANBarcode[code=EAN-8,ocrb=false,add_control]{2012345}
478
479 
```

(End definition for *\EANBarcode*. This function is documented on page 3.)

\GSSetup

```

480 {*package}
481 \NewDocumentCommand \GSSetup
482   { m }
483   { \keys_set:nn { GS1 } { #1 } }
484 
```

The following test files are used for this code: *GSSetup.tex*.

```

485 {*test & GSSetup}
486 After \verb|\GSSetup[ocrb=false,add_control]|:
487     \GSSetup[ocrb=false,add_control]
488
489 \verb|\EANBarcode[code=EAN-8]{2012345}|:
490     \EANBarcode[code=EAN-8]{2012345}
491
492 \verb|\EANBarcode[code=EAN-8,ocrb,add_control=false]{20123451}|:
493     \EANBarcode[code=EAN-8,ocrb,add_control=false]{20123451}
494
495 
```

(End definition for *\GSSetup*. This function is documented on page 4.)

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 | |
|---------------------------------------|---|
| \\" | <i>97, 98, 105,</i> <i>106, 107, 186, 201, 383, 385, 387, 388</i> |
| B | |
| \begin | 381 |
| bool commands: | |
| \bool_if:NTF | <i>278, 411, 431</i> |
| \bool_if:nTF | <i>404</i> |
| \bool_if_exist:NTF | <i>302</i> |
| \bool_new:N | <i>302</i> |
| \bool_set_eq:NN | <i>303</i> |
| \bool_set_false:N | <i>280</i> |
| \bool_set_true:N | <i>294</i> |
| \bool_until_do:nn | <i>209</i> |
| \c_false_bool | <i>5, 314</i> |
| \c_true_bool | <i>5, 309</i> |
| box commands: | |
| \box_ht:N | <i>286</i> |
| \box_scale:Nnn | <i>455</i> |
| \box_use:N | <i>458</i> |
| \box_wd:N | <i>433</i> |
| \l_tmpa_box | <i>281, 286, 432, 433, 454, 455, 458</i> |
| C | |
| control commands: | |
| \l_control_int | <i>231, 232, 234, 236, 239</i> |
| \l_control_tl | <i>228, 230, 233, 234</i> |
| cs commands: | |
| \cs_generate_variant:Nn | <i>317</i> |
| \cs_new_nopar:Npn | <i>8, 59, 112,</i> <i>166, 205, 246, 274, 300, 307, 312, 320</i> |
| D | |
| dim commands: | |
| \dim_compare:nNnTF | <i>398, 439</i> |
| \dim_const:Nn | <i>55, 56</i> |
| \dim_eval:n | <i>284, 287, 290, 291</i> |
| \dim_set:Nn | <i>401, 402, 433, 436, 447</i> |
| \dim_use:N | <i>445, 446</i> |
| \l_tmpa_dim | <i>436, 440</i> |
| \c_zero_dim | <i>73, 398</i> |
| E | |
| \EANBarcode | <i>2, 3, 392, 489, 490, 492, 493</i> |
| \EANControlDigit | <i>3, 369</i> |
| \end | <i>390</i> |
| \ExplFileVersion | <i>4</i> |
| \ExplSyntaxOff | <i>163, 202, 243, 271</i> |
| \ExplSyntaxOn | <i>128, 183, 226, 254</i> |
| F | |
| \fontsize | <i>406</i> |
| fp commands: | |
| \fp_to_decimal:N | <i>437</i> |
| \fp_to_int:n | <i>450, 456, 457</i> |
| G | |
| group commands: | |
| \group_begin: | <i>372, 396</i> |
| \group_end: | <i>377, 459</i> |
| GS commands: | |
| \l_GS_add_control_bool | <i>58, 411</i> |
| \l_GS_code_size_int | <i>2,</i> <i>4, 7, 58, 324, 340, 353, 400, 419, 426</i> |
| \l_GS_code_type_tl | <i>4, 7, 58</i> |
| \GS_cut_EAN_control_digit:N | <i>2, 165, 248, 374</i> |
| \l_GS_module_ht_dim | <i>5, 58, 287,</i> <i>337, 346, 350, 358, 362, 398, 401, 402</i> |
| \l_GS_module_wd_dim | <i>5, 58, 286, 290, 295, 326, 330,</i> <i>337, 350, 362, 364, 433, 437, 445, 447</i> |
| \l_GS_scale_fp | <i>58, 437, 450, 456, 457</i> |
| \l_GS_scale_to_font_bool | <i>58, 431</i> |
| \GS_set_code_digit_seq:Nn | <i>2, 111, 185, 187, 229, 256, 373, 409</i> |
| \GS_set_EAN_control_digit:N | <i>2, 245, 413</i> |
| \l_GS_system_seq | <i>327, 332, 344</i> |
| \GS_use_as_EAN_barcode:N | <i>2, 319, 454</i> |
| \l_GS_use_ocrb_bool | <i>58, 404</i> |
| GS internal commands: | |
| \c__GS_{sequence name}_seq | <i>4</i> |
| \c__GS_A0_seq | <i>27</i> |
| \c__GS_A1_seq | <i>27</i> |
| \c__GS_A2_seq | <i>27</i> |
| \c__GS_A3_seq | <i>27</i> |
| \c__GS_A4_seq | <i>27</i> |
| \c__GS_A5_seq | <i>27</i> |
| \c__GS_A6_seq | <i>27</i> |
| \c__GS_A7_seq | <i>27</i> |
| \c__GS_A8_seq | <i>27</i> |

| | | | | |
|---|--|--|--|--|
| \c__GS_A9_seq | <u>27</u> | \hbox_to_wd:nn | <u>295</u> , <u>326</u> , <u>330</u> , <u>364</u> | |
| \c__GS_AB0_seq | <u>15</u> | \hline | <u>382</u> , <u>384</u> , <u>389</u> | |
| \c__GS_AB1_seq | <u>15</u> | I | | |
| \c__GS_AB2_seq | <u>15</u> | \IfNoValueF | <u>397</u> | |
| \c__GS_AB3_seq | <u>15</u> | int commands: | | |
| \c__GS_AB4_seq | <u>15</u> | \int_add:Nn | <u>216</u> , <u>219</u> | |
| \c__GS_AB5_seq | <u>15</u> | \int_case:nnTF | <u>168</u> | |
| \c__GS_AB6_seq | <u>15</u> | \int_compare:nNnTF | | |
| \c__GS_AB7_seq | <u>15</u> | | <u>234</u> , <u>263</u> , <u>324</u> , <u>400</u> , <u>416</u> | |
| \c__GS_AB8_seq | <u>15</u> | \int_div_truncate:nn | <u>340</u> , <u>353</u> | |
| \c__GS_AB9_seq | <u>15</u> | \int_if_even:nTF | <u>213</u> | |
| \c__GS_B0_seq | <u>27</u> | \int_if_exist:NTF | <u>63</u> | |
| \c__GS_B1_seq | <u>27</u> | \int_mod:nn | <u>222</u> | |
| \c__GS_B2_seq | <u>27</u> | \int_new:N | <u>63</u> , <u>231</u> | |
| \c__GS_B3_seq | <u>27</u> | \int_set:NN | <u>64</u> , <u>222</u> | |
| \c__GS_B4_seq | <u>27</u> | \int_set_to_EAN_control_digit:NN | | |
| \c__GS_B5_seq | <u>27</u> | | <u>2</u> , <u>204</u> , <u>249</u> , <u>375</u> | |
| \c__GS_B6_seq | <u>27</u> | \int_step_inline:nnnn | <u>339</u> , <u>352</u> | |
| \c__GS_B7_seq | <u>27</u> | \int_to_arabic:n | <u>376</u> | |
| \c__GS_B8_seq | <u>27</u> | \int_use:N | <u>236</u> , <u>239</u> | |
| \c__GS_B9_seq | <u>27</u> | \int_while_do:nNnn | <u>423</u> | |
| \l__GS_black_bool | | \int_zero:N | <u>207</u> | |
| | <u>5</u> , <u>278</u> , <u>280</u> , <u>294</u> , <u>302</u> , <u>303</u> | \l_tmpa_int | <u>249</u> , <u>250</u> , <u>375</u> , <u>376</u> | |
| \l__GS_code_seq | | K | | |
| | <u>88</u> , <u>322</u> , <u>331</u> , <u>342</u> , <u>355</u> , <u>373</u> , <u>374</u> , <u>375</u> , <u>409</u> , <u>413</u> , <u>417</u> , <u>424</u> , <u>428</u> , <u>454</u> | keys commands: | | |
| \c__GS_extra_margin_seq | <u>27</u> | \keys_define:nn | <u>66</u> | |
| \c__GS_extra_separator_seq | <u>27</u> | \keys_set:nn | <u>397</u> , <u>483</u> | |
| \c__GS_margin_seq | <u>27</u> , <u>336</u> , <u>361</u> | M | | |
| \c__GS_module_min_width_dim | | msg commands: | | |
| | <u>54</u> , <u>442</u> , <u>446</u> , <u>449</u> | \msg_error:nnn | <u>177</u> , <u>240</u> , <u>269</u> , <u>421</u> | |
| \c__GS_module_norm_width_dim | <u>54</u> , <u>71</u> | \msg_fatal:nnn | <u>145</u> , <u>159</u> , <u>197</u> | |
| __GS_modules:Nn | <u>5</u> , <u>273</u> , <u>345</u> , <u>357</u> | \msg_line_context: | <u>94</u> , <u>105</u> | |
| __GS_modules:NnN | <u>5</u> , <u>273</u> | \msg_new:nnn | <u>103</u> | |
| __GS_modules_start_black:Nn | | \msg_new:nnnn | <u>92</u> | |
| | <u>5</u> , <u>273</u> , <u>336</u> , <u>361</u> | \msg_warning:nnnn | <u>444</u> | |
| __GS_modules_start_white:Nn | | N | | |
| | <u>5</u> , <u>273</u> , <u>349</u> | \NewDocumentCommand | <u>369</u> , <u>393</u> , <u>481</u> | |
| __GS_new_seq_c:Nn | | P | | |
| | <u>4</u> , <u>7</u> , <u>16</u> , <u>17</u> , <u>18</u> , <u>19</u> , <u>20</u> , <u>21</u> , <u>22</u> , <u>23</u> , <u>24</u> , <u>25</u> , <u>28</u> , <u>29</u> , <u>30</u> , <u>31</u> , <u>32</u> , <u>33</u> , <u>34</u> , <u>35</u> , <u>36</u> , <u>37</u> , <u>38</u> , <u>39</u> , <u>40</u> , <u>41</u> , <u>42</u> , <u>43</u> , <u>44</u> , <u>45</u> , <u>46</u> , <u>47</u> , <u>48</u> , <u>49</u> , <u>50</u> , <u>51</u> , <u>52</u> | \ProvidesExplPackage | <u>3</u> | |
| \c__GS_separator_seq | <u>27</u> , <u>349</u> | R | | |
| __GS_set_key_code:nn | | \raggedright | <u>182</u> , <u>463</u> | |
| | <u>4</u> , <u>58</u> , <u>75</u> , <u>76</u> , <u>77</u> , <u>78</u> | \RequirePackage | <u>5</u> , <u>368</u> | |
| \c__GS_special_seq | <u>27</u> | rule commands: | | |
| \GSSetup | <u>3</u> , <u>4</u> , <u>480</u> | \rule:nnn | <u>282</u> | |
| H | | | | |
| hbox commands: | | S | | |
| \hbox_overlap_left:n | <u>333</u> | \selectfont | <u>406</u> | |
| \hbox_overlap_right:n | <u>343</u> , <u>356</u> | seq commands: | | |
| \hbox_set:Nn | <u>281</u> , <u>432</u> , <u>454</u> | \c_empty_seq | <u>114</u> | |

| | | | |
|---|----------------------------|--|--|
| \seq_count:N | 169, 214, 417, 424 | \t_testa_t1 | 268 |
| \seq_get_right>NN | 261 | testb commands: | |
| \seq_gset_eq>NN | 12 | \l_testb_seq | 130, |
| \seq_if_empty_p:N | 210 | 136, 139, 150, 153, 158, 187, 191, 196 | |
| \seq_map_inline:Nn | 137, 151, 189, 276 | \l_testb_t1 | 257, 258, 262, 263 |
| \seq_new:N | | tl commands: | |
| .. 10, 89, 129, 130, 184, 227, 255, 257 | | \tl_if_eq:NNTF | 141, 155, 193 |
| \seq_pop_left>NN | | \tl_if_exist:NTF | 61 |
| 139, 153, 191, 212, 331, 342, 344, 355 | | \tl_if_in:nnTF | 118 |
| \seq_pop_right>NN | 172, 174, 230, 258 | \tl_map_inline:nn | 115 |
| \seq_put_left:Nn | 428 | \tl_new:N | 61, 228, 260 |
| \seq_put_right:Nn | 122, 250 | \tl_set:Nn | 62, 140, 154, 192 |
| \seq_set_eq>NN | 114, 208, 322, 327, 332 | \tl_show:N | 143, 144, 157, 158, 195, 196 |
| \seq_set_split:Nnn | 11, 150 | \tl_use:N | 233, 262, 265, 268 |
| \seq_use:Nnnn | 134, 186, 201 | \l_tmpa_t1 | |
| \l_tmpa_seq | 11, 12, 208, 210, 212, 214 | .. 139, 141, 153, 155, 191, 193, 331, | |
| | | 332, 333, 342, 343, 345, 355, 356, 357 | |
| T | | \l_tmpb_t1 | 140, 141, 154, |
| tempa commands: | | 155, 192, 193, 212, 216, 219, 344, 345 | |
| \l_tempa_t1 | 172, 174 | | |
| testa commands: | | U | |
| \l_testa_seq | 129, | \usefont | 406 |
| 132, 134, 137, 149, 151, 157, 184, | | | |
| 185, 186, 188, 189, 195, 201, 227, | | V | |
| 229, 230, 232, 255, 256, 258, 259, 261 | | \verb | 464, 467, 470, 473, 476, 486, 489, 492 |
| \l_testa_t1 | 260, 261, 263, 265 | | |