tiaoma

A barcode generator for typst that provides type safe API bindings for Zint (GitHub) library through a WASM plugin.

See official Zint manual for a more in-depth description of supported functionality.

API

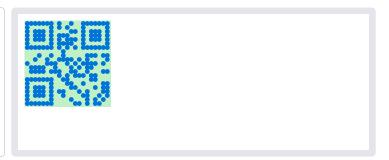
Some generators require additional configuration (such as composite codes), this can be achieved by passing <u>options</u> to Zint.

barcode

Draw a barcode SVG of any supported symbology.

Example:

```
#tiaoma.barcode("12345678", "QRCode", options: (
    scale: 2.0,
    fg-color: blue,
    bg-color: green.lighten(70%),
    output-options: (
        barcode-dotty-mode: true
    ),
    dot-size: 1.2,
))
```



- data (str): Data to encode.
- symbology (str): Symbology type name; must be one of <u>supported types</u>.

```
Example values: "Code11", "C25Standard", ...
```

• options (dictionary): Additional options to pass to Zint.

See the <u>configuration section</u> for details on available options and how to use them.

• ..args (any): Any additional arguments to forward to image.decode function.

Parameters

```
barcode(
  data,
  symbology,
  options,
   ..args
) -> content
```

dm-size

Returns int option value for given Data Matrix width and height.

Zint allows square and rectangular values to be enforced with DM_SQUARE and DM_DMRE Option 3 values.

- width (int): Data Matrix width
- height (int): Data Matrix height

Parameters

```
dm-size(
  height,
  width
) -> int
```

Shortcut functions

Most barcodes are supported through <u>shortcut functions</u>. They accept the same arguments as barcode function but don't require symbology to be specified.

Zint configuration

All exported functions support optionally providing the options dictionary which is passed to Zint. This provides means to fully configure generated images.

The following values are valid for the ${\tt options}$ dictionary:

Field	Type	Description	Default
height	float	Barcode height in X-dimensions (ignored for fixed-width barcodes)	none
scale	float	Scale factor when printing barcode, i.e. adjusts X-dimension	1.0
whitespace-width	int	Width in X-dimensions of whitespace to left & right of barcode	0
whitespace-height	int	Height in X-dimensions of whitespace above & below the barcode	0
border-width	int	Size of border in X-dimensions	0
output-options	<pre>int or array or dictionary</pre>	Various output parameters (bind, box etc, see below)	0
fg-color	color	foreground color	black
bg-color	color	background color	white
primary	str	Primary message data (MaxiCode, Composite)	пп
option-1	int	Symbol-specific options (see <u>manual</u>)	-1
option-2	int	Symbol-specific options (see <u>manual</u>)	0
option-3	int or str	Symbol-specific options (see <u>manual</u>)	0
show-hrt	bool	Whether to show Human Readable Text (HRT)	true
input-mode	int or string or array or dictionary	Encoding of input data	0
eci	int	Extended Channel Interpretation.	0
dot-size	float	Size of dots used in BARCODE_DOTTY_MODE.	4.0 / 5.0
text-gap	float	Gap between barcode and text (HRT) in X-dimensions.	1.0
guard-descent	float	Height in X-dimensions that EAN/UPC guard bars descend.	5.0

Input Mode

Input mode options allow specifying how Zint should handle input data. Zint uses int bitflags for these, but tiaoma allows you to specify them using several other formats as documented below.

The following options are supported:

Input format (mutually exclusive)					
Constant	int	str	Description		
DATA_MODE	0	"data"	Use full 8-bit range interpreted as binary data.		
UNICODE_MODE	1	"unicode"	Use UTF-8 input.		
GS1_MODE	2	"gs1"	Encode GS1 data using FNC1 characters.		

Behavior customization					
Constant	int	str	Description		
ESCAPE_MODE	8	"escape"	Process input data for escape sequences.		
GS1PARENS_MODE	16	"gs1-parentheses"	Parentheses (round brackets) used in GS1 data instead of square brackets to delimit Application Identifiers (parentheses must not otherwise occur in the data).		
GS1NOCHECK_MODE	32	"gsl-no-check"	Do not check GS1 data for validity, i.e. suppress checks for valid AIs and data lengths. Invalid characters (e.g. control characters, extended ASCII characters) are still checked for.		
HEIGHTPERROW_MODE	64	"height-per-row"	Interpret the height variable as per-row rather than as overall height.		
FAST_MODE	128	"fast"	Use faster if less optimal encodation for symbologies that support it (currently Data Matrix only).		
EXTRA_ESCAPE_MODE	256	"extra-escape"	Undocumented.		

String Value

input_mode of str type is assumed to be a *input format* value from the first table.

Array Value

input_mode of array type is assumed to be a list of str values from the above tables; individual constants will be converted to ints and unioned together.

Dictionary Value

input_mode of dictionary type is assumed to be str-bool pairs where keys are constants from the above table.

Additionally, *input format* can be specified as a str value paired to "format" key.

In other words, columns of the following table are equivalent:

dictionary	array	str	int
("format": "data")	("data")	"data"	0
("unicode": true)	("unicode")	"unicode"	1
("gs1": true, "gs1-no- check": true)	("gs1", "gs1-no-check")	N/A	34

Output Options

Output options allow specifying how Zint should generate the barcode/symbol.

Constant	int	str	Description
BARCODE_BIND_TOP	1	"barcode-bind-top"	Boundary bar <i>above</i> the symbol and between rows if stacking multiple symbols.
BARCODE_BIND	2	"barcode-bind"	Boundary bars <i>above</i> and <i>below</i> the symbol and between rows if stacking multiple symbols.

BARCODE_BOX	4	"barcode-box"	Add a box surrounding the symbol and whitespace.
SMALL_TEXT	32	"small-text"	Use a smaller font for the Human Readable Text.
BOLD_TEXT	64	"bold-text"	Embolden the Human Readable Text.
CMYK_COLOUR	128	"cmyk-color"	Select the CMYK colour space option for Encapsulated PostScript and TIF files.
BARCODE_DOTTY_MODE	256	"barcode-dotty-mode"	Plot a matrix symbol using dots rather than squares.
GS1_GS_SEPARATOR	512	"gs1-gs-separator"	Use GS instead of FNC1 as GS1 separator (Data Matrix only).
BARCODE_QUIET_ZONES	2048	"barcode-quiet-zones"	Add compliant quiet zones (additional to any specified whitespace).
BARCODE_NO_QUIET_ZONES	4096	"barcode-no-quiet-zones"	Disable quiet zones, notably those with defaults.
COMPLIANT_HEIGHT	8192	"compliant-height"	Warn if height not compliant and use standard height (if any) as default.
EANUPC_GUARD_WHITESPACE	16384	"ean-upc-guard-whitespace"	Add quiet zone indicators ("<" / ">") to HRT whitespace (EAN/UPC)
EMBED_VECTOR_FONT	32768	"embed-vector-font"	Embed font in vector output.

Array Value

output_options of array type assumed to be str values from the above table.

Dictionary Value

output_options of dictionary type assumed to be str-bool pairs where keys are constants from the above table.

Option 3

As there's constants associated with option_3 values, this package allows specifying the value as either an int or a str.

The following table documents supported values and their str representations:

Constant	int	str	Description
DM_SQUARE	100	"square"	Only consider square versions on automatic symbol size selection
DM_DMRE	101	"rect"	Consider DMRE versions on automatic symbol size selection
DM_ISO_144	128	"iso-144"	Use ISO instead of "de facto" format for 144x144 (i.e. don't skew ECC)
ZINT_FULL_MULTIBYTE	200	"full-multibyte"	Enable Kanji/Hanzi compression for Latin-1 & binary data
ULTRA_COMPRESSION	128	"compression"	Enable Ultracode compression (experimental)

Examples

EAN (European Article Number)

7.4377	(Table 1)
EANX Example:	EAN-14 Example:
#eanx("1234567890")	#ean14("1234567890")
Result:	Result:
0 012345 678905	(01)00012345678905
EAN-13	EAN-8
Example:	Example:
#eanx("6975004310001")	#eanx("12345564")
Result:	Result:
6 975004 310001	0 000123 455640
EAN-5	EAN-2
Example:	Example:
#eanx("12345")	#eanx("12")
Result:	Result:

PDF417

Micro PDF417
Example:

#micro-pdf417("1234567890")

Result:

Compact PDF417 Example:

#pdf417-comp("1234567890")

Result:

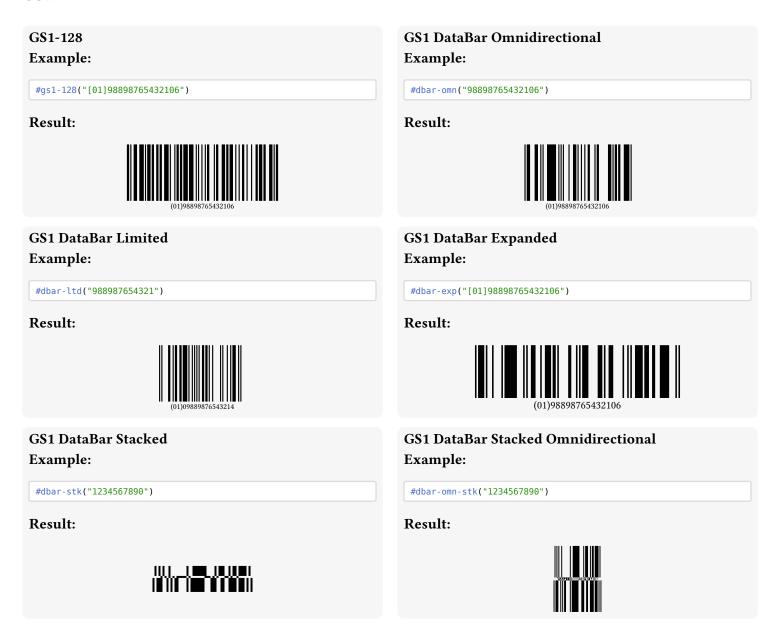


PDF417 Example:

#pdf417("1234567890")

Result:





Zint supports Omnidirectional, Limited, Expanded, Stacked and Composite Code variants of GS1. See <u>configuration</u> section for information on how to use them.

Standard

Example:

#c25-standard("123")

Result:



IATA

Example:

#c25-iata("1234567890")

Result:



Industrial

Example:

#c25-ind("1234567890")

Result:



Interleaved

Example:

#c25-inter("1234567890")

Result:



Data Logic

Example:

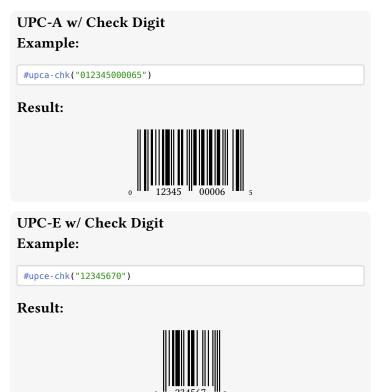
#c25-logic("1234567890")

Result:



UPC (Universal Product Code)

UPC-A Example: #upca("01234500006") Result: UPC-E Example: #upce("123456") Result:



HIBC (Health Industry Barcodes)

Code 128 Example: #hibc-128("1234567890") **Result: Data Matrix** QR **Example:** #hibc-dm("1234567890") **Result: PDF417 Example:** #hibc-pdf("1234567890") **Result:** Codablock-F **Example:** #hibc-codablock-f("1234567890") **Result: Result:**









Postal

Australia Post Redirection Example:

#aus-redirect("12345678")

Result:

ես<u>իլիկիկուրի</u>կուգիրեւկեն

Australia Post Reply Paid Example:

#aus-reply("12345678")

Result:

եսոլիկորոլլեր, երերի և

Australia Post Routing Example:

#aus-route("12345678")

Result:

 $\{\{\{\{i,j\}\}\}\}_{j=1}^{n},\{\{j\}\}_$

Australia Post Standard Customer Example:

#aus-post("12345678")

Result:

Brazilian CEPNet Postal Code Example:

#cepnet("1234567890")

Result:

laalladaladladaddalladaalladdallaaddaadda

DAFT Code Example:

#daft("DAFTFDATATFDTFAD")

Result:

լդիկեսիրկ

Deutsche Post Identcode

Example:

#dp-ident("1234567890")

Result:



Deutsche Post Leitcode Example:

#dp-leitcode("1234567890")

Result:



Deutsher Paket Dienst

Example:

#dpd("0123456789012345678901234567")

Result:



Dutch Post KIX Code

Example:

#kix("1234567890")

Result:

դկդիրկիկիկիկիկիկիկի

Japanese Postal Code Example:

#japan-post("1234567890")

Result:

<u>իլիիլիկիկիլիկիկիկիվիստետերերերերերերերերել</u>

POSTNET

Example:

#postnet("1234567890")

Result:



Korea Post

Example:

#korea-post("123456")

Result:



Royal Mail 2D Mailmark (CMDM)

Example:

```
#mailmark-2d(
    32, 32,
    "JGB 011123456712345678CW14NJ1T 0EC2M2QS
REFERENCE1234567890QWERTYUIOPASDFGHJKLZXCVBNM"
)
```

Result:



Royal Mail 4-State Customer Code Example:

#rm4scc("1234567890")

Result:

<u> Երկել Արժիր Արևելի ի ինի հիշվի ինի</u>

Royal Mail 4-State Mailmark Example:

#mailmark-4s("21B2254800659JW509QA6Y")

Result:



Universal Postal Union S10 Example:

#upus10("RR072705659PL")

Result:



UPNQR (Univerzalnega Plačilnega Naloga QR) Example:

#upnqr("1234567890")

Result:



USPS Intelligent Mail Example:

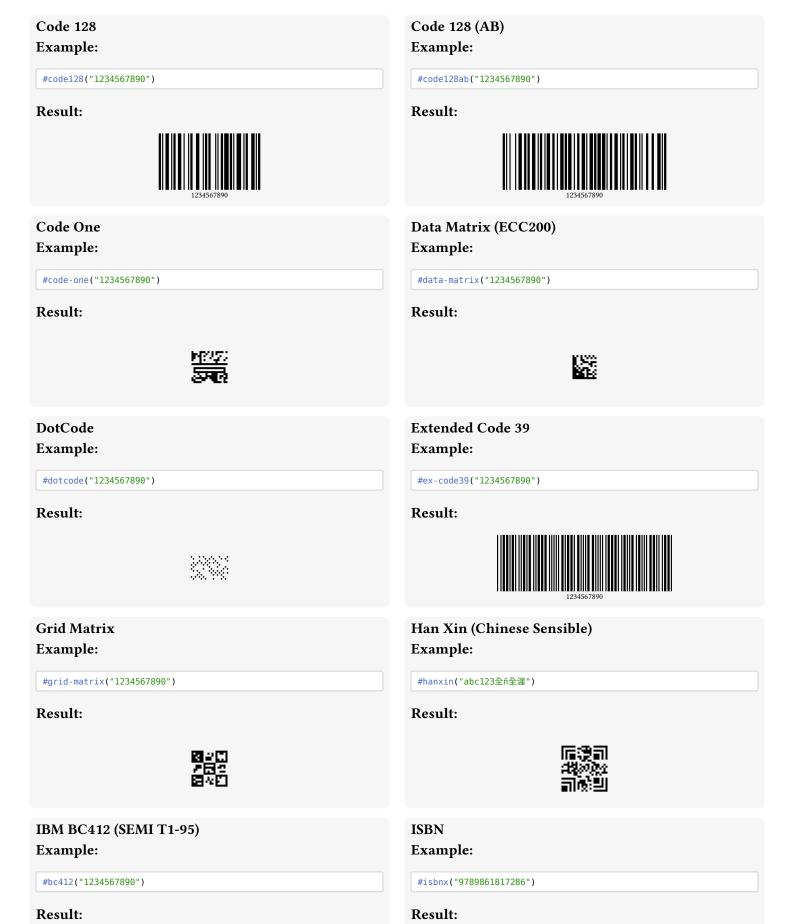
#usps-imail("01300123456123456789")

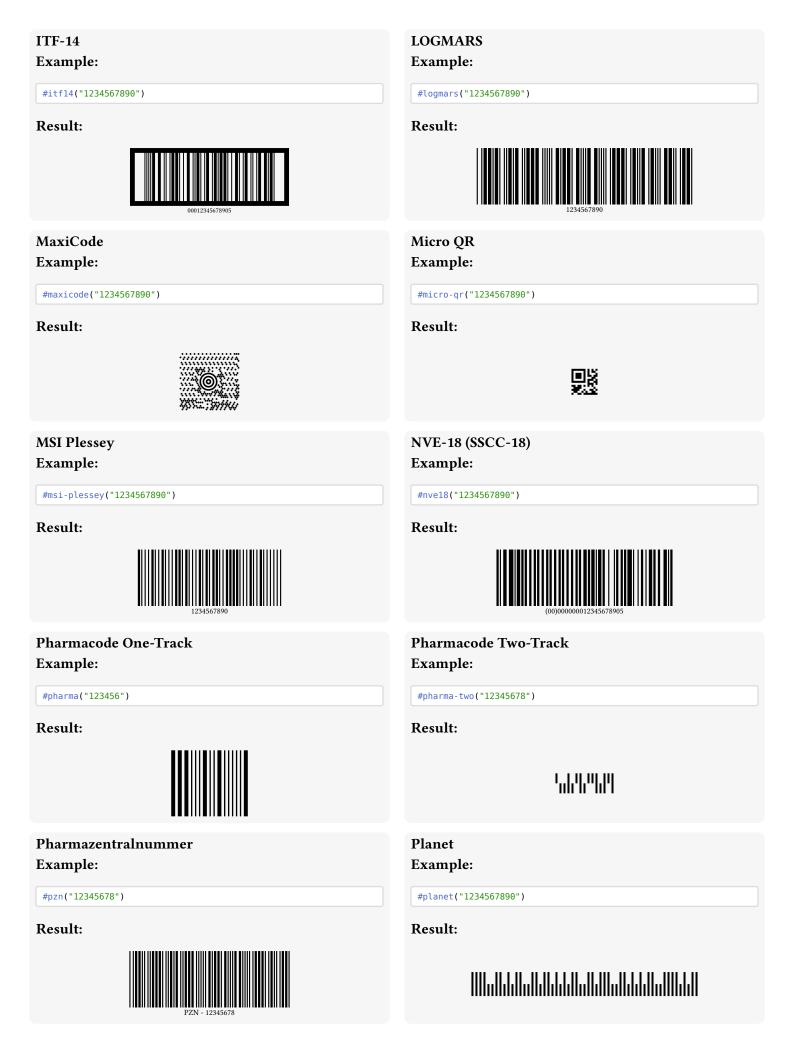
Result:

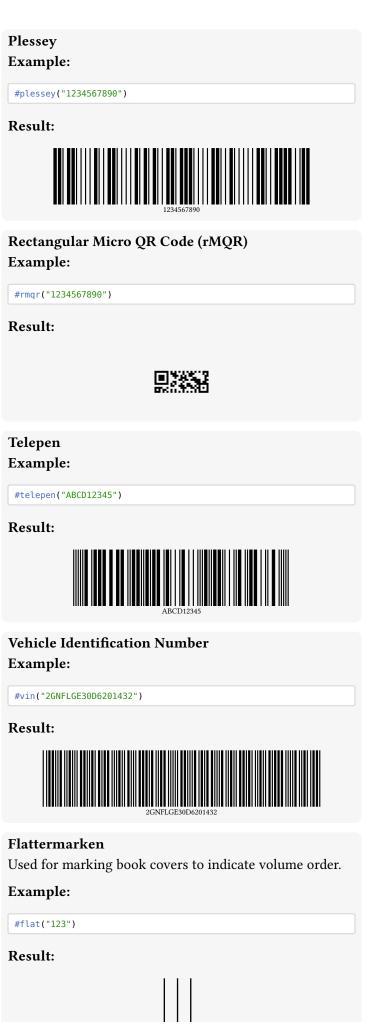
հուժկվակերկից հերկանկեր հերկանին հերկանին հերկանին հերկանին հերկանին հերկանին հերկանին հերկանին հերկանին հերկա

Other Generic Codes

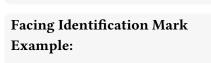
Aztec Code Aztec Rune Example: Example: #aztec("1234567890") #azrune("122") **Result: Result:** 鷆 回 **Channel Code** Codabar **Example: Example:** #channel("123456") #codabar("A123456789B") **Result: Result:** Codablock-F Code 11 **Example: Example:** #codablock-f("1234567890") #code11("0123452") **Result: Result:** Code 16k Code 32 **Example: Example:** #code32("12345678") #code16k("1234567890") **Result: Result:** Code 39 Code 49 **Example: Example:** #code39("1234567890") #code49("1234567890") **Result: Result:**











Result:

#fim("A")



Symbology ValuesFollowing symbology values are supported:

"Code11" "Code39" "Code128" "Flat" "UPCAChk" "Logmars"	"C25Standard" "ExCode39" "DPLEIT" "DBar0mn" "UPCE" "Pharma"	"C25Inter" "EANX" "DPIDENT" "DBarLtd" "UPCEChk" "PZN"	"C25IATA" "EANXChk" "Code16k" "DBarExp" "Postnet" "PharmaTwo"	"C25Logic" "GS1128" "Code49" "Telepen" "MSIPlessey" "CEPNet"	"C25Ind" "Codabar" "Code93" "UPCA" "FIM" "PDF417"
"PDF417Comp" "AusRoute" "VIN" "DBarOmnStk" "TelepenNum" "MicroQR" "HIBCMicPDF" "UPUS10" "DBarOmnCC" "DBarOmnStkCC"	"MaxiCode" "AusRedirect" "CodablockF" "DBarExpStk" "ITF14" "HIBC128" "HIBCCodablockF" "Mailmark4S" "DBarLtdCC" "DBarExpStkCC" "RMQR"	"QRCode" "ISBNX" "NVE18" "Planet" "KIX" "HIBC39" "HIBCAztec" "AzRune" "DBarExpCC" "Channel" "BC412"	"Code128AB" "RM4SCC" "JapanPost" "MicroPDF417" "Aztec" "HIBCDM" "DotCode" "Code32" "UPCACC" "Code0ne"	"AusPost" "DataMatrix" "KoreaPost" "USPSIMail" "DAFT" "HIBCQR" "HanXin" "EANXCC" "UPCECC" "GridMatrix"	"AusReply" "EAN14" "DBarStk" "Plessey" "DPD" "HIBCPDF" "Mailmark2D" "GS1128CC" "DBarStkCC" "UPNQR"