

# The LUA-PHYSICAL library

Version 0.1

T. Jenni

May 2, 2019

## Abstract

`lua-physical` is a pure Lua library which provides functions and object for doing computation with physical quantities. It has been written to simplify the creation problem sets. The package provides units of the SI and the imperial system. Furthermore an almost complete set of international currencies are supported, however without online exchange rates. In order to display the numbers with measurement uncertainties, the package is able to perform gaussian error propagation.

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Loading</b>	<b>3</b>
2.1	Dependencies . . . . .	4
<b>3</b>	<b>Usage</b>	<b>4</b>
3.1	Unit conversion . . . . .	4
3.1.1	Temperature Conversion . . . . .	5
3.1.2	Uncertainty . . . . .	6
<b>4</b>	<b>Supported Units</b>	<b>8</b>
4.1	Base Units . . . . .	8
4.2	Constants . . . . .	10
4.3	Coherent derived units in the SI . . . . .	11
4.4	Non-SI units accepted for use with the SI . . . . .	12
4.5	Nominal Astronomical Units . . . . .	12
4.6	Other Non-SI units . . . . .	13
4.7	Imperial Units . . . . .	15
4.8	U.S. customary units . . . . .	17
4.9	International Currencies . . . . .	18
4.9.1	Pegged International Currencies . . . . .	26

<b>5</b>	<b>Lua Documentation</b>	<b>30</b>
5.1	physical.Quantity . . . . .	30
5.2	physical.Dimension . . . . .	38
5.3	physical.Unit . . . . .	39
5.4	physical.Number . . . . .	40
	<b>Bibliography</b>	<b>43</b>
	<b>Index</b>	<b>44</b>
	<b>Index of Currencies</b>	<b>47</b>

# 1 Introduction

The author of this package is a teacher at the high school *Kantonsschule Zug* in Switzerland. The main use of this package is to write physics problem sets. Lua<sup>1</sup>TeX does make it possible to integrate physical calculations directly. The package has been in use since 2016. Many bugs have been found and fixed. Nevertheless it still is possible, that some were not found yet. Therefore the author recommends not to use this package industry or science. If one does so, it's the responsibility of the user to check results for plausability. If the user finds some bugs, they can be reported at [github.com](https://github.com) or directly to the author (`thomas.jenni` (at) `ksz.ch`).

# 2 Loading

This package is a pure Lua library. Therefore one has to require it explicitly by calling `require("physical")`. For printing results, the `siunitx` package can be used. It's recommended to define a macro like `\q` to convert the lua quantity object to a `siunitx` expression.

The following Latex preambel loads the `lua-physical` package and creates a macro `\q` for printing physical quantities.

Listing 1: basic preamble

```
1  \usepackage{lua-physical}
2  \usepackage{siunitx}
3
4  % configure siunitx
5  \sisetup{
6    output-decimal-marker = {.,},
7    per-mode = symbol,
8    separate-uncertainty = true,
9    add-decimal-zero = true,
10   exponent-product = \cdot,
11   round-mode = off
12 }
13
14 % load the lua-physical package
15 \begin{luacode*}
16   physical = require("physical")
17   N = physical.Number
18 \end{luacode*}
19
20 % print a physical quantity
21 \newcommand{\q}[1]{%
22   \directlua{tex.print(physical.Quantity.tosiunitx(#1,"scientific-
23     notation=fixed,exponent-to-prefix=false"))}%
24 }
```

## 2.1 Dependencies

This is a standalone package. If a pretty print to Lua<sup>A</sup>T<sub>E</sub>X is needed, the package `siunitx` should be installed.

## 3 Usage

Given the basic preamble, units can be used in lua code directly. By convention, all units have an underscore in front of them, i.e. meter is `_m`, second is `_s`. All available units are listed in chapter 4. The Result of the calculation can be printed to Lua<sup>A</sup>T<sub>E</sub>X by using the macro `\q{}`.

Listing 2: The velocity of a car.

```
1  \begin{luacode}
2    s = 10 * _m
3    t = 2 * _s
4    v = s/t
5  \end{luacode}
6
7  A car travels $\q{s}$ in $\q{t}$. calculate its velocity.
8  $$
9    v=\frac{s}{t} = \frac{\q{s}}{\q{t}} = \q{v}
10  $$
```

A car travels 10 m in 2 s. Calculate its velocity.

$$v = \frac{s}{t} = \frac{10 \text{ m}}{2 \text{ s}} = 5.0 \text{ m/s}$$

In the above listing 2, the variable `s` stands for displacement and has the unit meter `_m`. The variable `t` stands for time and is given in seconds `_s`. If mathematical operations are done on them, new physical quantities are created. In the problem above, the velocity `v` is calculated by dividing `s` by `t`. The instance `v` has the derived unit m/s. By using the macro `\q{}` all quantities can be printed to the Lua<sup>A</sup>T<sub>E</sub>X code directly.

### 3.1 Unit conversion

Very often, the result of a calculation has to be converted to another unit. In the following listing 3 the task is to calculate the volume of a cuboid with lengths given in three different units. If the volume is calculated by multiplying all three lengths, the unit of the result is cm mm m. If the unit cm<sup>3</sup> is preferred, it has to be converted explicitly. The conversion function is called `to()` and is available on all physical quantity instances. At first this looks a bit cumbersome. The reason of this behaviour is, that the software is not able to guess the unit of the result. In many cases, like in the example here, it's not clear what unit the result should have. Therefore the user has always to give the target unit explicitly.

Listing 3: The volume of a cuboid.

```

1 \begin{luacode}
2   a = 12 * _cm
3   b = 150 * _mm
4   c = 1.5 * _m
5
6   V = a*b*c
7 \end{luacode}
8
9 Find the volume of a rectangular cuboid with lengths  $\text{\textbackslash q{a}}$ %,
10  $\text{\textbackslash q{b}}$ % and  $\text{\textbackslash q{c}}$ %.
11 $$
12   V= a \text{\textbackslash cdot} b \text{\textbackslash cdot} c
13   = \text{\textbackslash q{a}} \text{\textbackslash cdot} \text{\textbackslash q{b}} \text{\textbackslash cdot} \text{\textbackslash q{c}}
14   = \text{\textbackslash q{V}}
15   = \text{\textbackslash uuline{\text{\textbackslash q{V:to}(\_dm^3)}}}
16 $$

```

Find the volume of a rectangular cuboid with lengths 12 cm, 150 mm and 1.5 m.

$$V = a \cdot b \cdot c = 12 \text{ cm} \cdot 150 \text{ mm} \cdot 1.5 \text{ m} = 2700.0 \text{ cm mm m} = \underline{\underline{27.0 \text{ dm}^3}}$$

### 3.1.1 Temperature Conversion

Most of the units transform linearly to base units. Exceptions are the unit degree Celsius `_degC` and degree Fahrenheit `_degF`. These units are ambiguous and can be interpreted as temperature differences or as an absolute temperatures. In the latter case, the conversion to base units is not a linear, but an affine transformation. This is because degree Celsius and degree Fahrenheit scales have their zero points at different temperatures compared to the unit Kelvin.

By default `_degC` and `_degF` units are standing for temperature differences. If one wants to have it converted absolutely, it has to be done adding / subtracting the quantities `_degC_0` and `_degF_0`.

In the following problem, listing 4 , the task is to convert temperatures given in the unit degree Celsius and degree Fahrenheit to Kelvin.

Listing 4: Temperature conversion.

```

1 \begin{luacode}
2   theta_1 = 110 * _degC
3   T_1 = ( theta_1 + _degC_0 ):to(_K)
4   theta_1 = T_1:to(_degC) - _degC_0
5
6   theta_2 = 212 * _degF
7   T_2 = ( theta_2 + _degF_0 ):to(_K)
8   theta_2 = T_2:to(_degF) - _degF_0
9 \end{luacode}
10
11 \begin{align*}
12   \text{\textbackslash vartheta}_1 \text{ \&= } \text{\textbackslash q{theta}_1} \text{ \&\&}
13   T_1 \text{ \&= } \text{\textbackslash q{T}_1} \text{ \&\&}
14   \%

```

```

15 \vartheta_2 &= \q{theta_2} \\
16 T_2 &= \q{T_2} \\
17 \end{align*}

```

$$\vartheta_1 = 110.0^\circ\text{C}$$

$$T_1 = 383.15\text{ K}$$

$$\vartheta_2 = 212.0^\circ\text{F}$$

$$T_2 = 373.15\text{ K}$$

### 3.1.2 Uncertainty

The package supports uncertainty propagation. To create a number with an uncertainty, an instance of `physical.Number` has to be created, see listing 5. It has to be remembered, that `N` is a alias for `physical.Number`. The first argument of the constructor `N(mean, uncertainty)` is the mean value and the second one the uncertainty of the measurement. If the proposed preamble 1 is used, the uncertainty is by default separated from the mean value by a plus-minus sign. For the uncertainty propagation the gaussian formula

$$\Delta f = \sqrt{\left(\frac{\partial f}{\partial x_1} \cdot \Delta x_1\right)^2 + \dots + \left(\frac{\partial f}{\partial x_n} \cdot \Delta x_n\right)^2}$$

is used. This formula is a good estimation for the uncertainty  $\Delta f$ , if the quantities  $x_1, \dots, x_n$  the function  $f$  depends on, have no correlation. Further, the function  $f$  has to change linear, if quantities  $x_i$  are changed in the range of their uncertainties.

Listing 5: Uncertainty in area calculation.

```

1 \begin{luacode}
2   a = N(2,0.1) * _m
3   b = N(3,0.1) * _m
4
5   A = (a*b):to(_m^2)
6 \end{luacode}
7
8 Calculate the area of a rectangle with lengths  $\q{a}$  and  $\q{b}$ .
9 $$
10  A = a \cdot b
11  = \q{a} \cdot \q{b}
12  = \uuline{\q{A}}
13  $$

```

Calculate the area of a rectangle with lengths  $(2.00 \pm 0.10)\text{ m}$  and  $(3.00 \pm 0.10)\text{ m}$ .

$$A = a \cdot b = (2.00 \pm 0.10)\text{ m} \cdot (3.00 \pm 0.10)\text{ m} = \underline{\underline{(6.0 \pm 0.4)\text{ m}^2}}$$

Instead of printing always the uncertainties, one can use the uncertainty calculation to provide significant digits.

In the following problem, listing 6 , the task is to calculate the volume of an ideal gas. Given are pressure  $p$  in `_bar`, amount of substance  $n$  in `_mol` and temperature  $T$  in degree celsius `_degC`. In order to do the calculation, one has to convert  $T$ , which is given as an absolute temperature in degree celsius to the base unit Kelvin first. By setting `N.omitUncertainty = true`, all uncertainties are not printed.

Listing 6: Volume of an ideal gas.

```

1  \begin{luacode}
2      N.omitUncertainty = true
3      p = N(1.013,0.0001) * _bar
4      n = N(1,0.01) * _mol
5      T = N(30,0.1) * _degC
6
7      V = ( n * _R * (T + _degC_0):to(_K) / p ):to(_L)
8  \end{luacode}
9
10 An ideal gas ( $\text{\textbackslash q{n}}$ ) has a pressure of  $\text{\textbackslash q{p}}$  and a temperature
    of  $\text{\textbackslash q{T}}$ . Calculate the volume of the gas.
11 $$
12 V=\frac{\text{\textbackslash q{n}} \cdot \text{\textbackslash q{R}} \cdot \text{\textbackslash q{(T + _degC_0):to(_K)}}}{\text{\textbackslash q{p}}}
13 = \text{\textbackslash q{V}}
14 = \underline{\text{\textbackslash q{V}}}
15 $$

```

An ideal gas (1.0 mol) has a pressure of 1.013 bar and a temperature of 30 °C. Calculate the volume of the gas.

$$V = \frac{1.0 \text{ mol} \cdot 8.31 \text{ J}/(\text{mol K}) \cdot 303 \text{ K}}{1.013 \text{ bar}} = \underline{\underline{25 \text{ L}}}$$

## 4 Supported Units

All supported units are listed in this chapter. Subchapter 4.1 lists the seven base units of the International System of Units (SI). In subchapter 4.2 mathematical and physical constants are defined. The subchapter 4.3 contains all coherent derived units from the SI system and 4.4 those which are accepted to use with the SI.

The subchapter 4.5 lists nominal astronomical units, which are proposed by [3]. Subchapter 4.6 lists units, which are common but outside of the SI system. The subchapters 4.7 and 4.8 are dedicated to imperial and U.S. customary units. The last subchapter 4.9 contains international currencies.

### 4.1 Base Units

Quantity	Unit	Symbol	Dim.	Definition
number	–	<code>_1</code>	1	The dimensionless number one.
time	second	<code>_s</code>	T	The SI unit of time. It is defined by taking the fixed numerical value of the caesium frequency $\Delta\nu_{Cs}$ , the unperturbed ground-state hyperfine transition frequency of the caesium 133 atom, to be 9 192 631 770 when expressed in the unit 1/s.
length	meter	<code>_m</code>	L	The SI unit of length. It is defined by taking the fixed numerical value of the speed of light in vacuum $c$ to be 299 792 458 when expressed in the unit of 1.0 m/s.
mass	kilogram	<code>_kg</code>	M	The SI unit of mass. It is defined by taking the fixed numerical value of the Planck constant $h$ to be $6.626\,070\,15 \cdot 10^{-34}$ when expressed in $\text{m}^2 \text{kg/s}$ .
electric current	ampere	<code>_A</code>	I	The SI unit of electric current. It is defined by taking the fixed numerical value of the elementary charge $e$ to be $1.602\,176\,634 \cdot 10^{-19}$ when expressed in A s.



Quantity	Unit	Symbol	Dim.	Definition
thermodynamic temperature	kelvin	<code>_K</code>	$\Theta$	The SI unit of the thermodynamic temperature. It is defined by taking the fixed numerical value of the Boltzmann constant $k_B$ to be $1.380\,649 \cdot 10^{-23}$ when expressed in $1.0 \text{ kg m}^2/(\text{s}^2 \text{ K})$
amount of substance	mole	<code>_mol</code>	N	The SI unit of amount of substance. One mole contains exactly $6.022\,140\,76 \cdot 10^{23}$ elementary entities. This number is the fixed numerical value of the Avogadro constant $N_A$ when expressed in $1/\text{mol}$ .
luminous intensity	candela	<code>_cd</code>	J	The SI unit of luminous intensity in a given direction. It is defined by taking the fixed numerical value of the luminous efficacy of monochromatic radiation of frequency $5.4 \cdot 10^{14} \text{ Hz}$ , $K_{cd}$ , to be 683 when expressed in the unit $\text{cd sr s}^3/(\text{kg m}^2)$ .
information	bit	<code>_bit</code>	B	The smallest amount of information.
currency	euro	<code>_EUR</code>	C	The value of the currency Euro.

## 4.2 Constants

All physical constants are taken from the NIST webpage [1].

Name	Symbol	Definition
pi	<code>_Pi</code>	$3.1415926535897932384626433832795028841971 * \_1$
eulersnumber	<code>_E</code>	$2.7182818284590452353602874713526624977572 * \_1$
speedoflight	<code>_c</code>	$299792458 * \_m/_s$
gravitationalconstant	<code>_Gc</code>	$N(6.67408e-11, 3.1e-15) * \_m^3/(\_kg*_s^2)$
planckconstant	<code>_h_P</code>	$6.62607015e-34 * \_J*_s$
reducedplanckconstant	<code>_h_Pbar</code>	$\_h\_P/(2*_Pi)$
elementarycharge	<code>_e</code>	$1.602176634e-19 * \_C$
vacuumpermeability	<code>_u_0</code>	$4e-7*_Pi * \_N/_A^2$
vacuumpermittivity	<code>_e_0</code>	$1/(\_u_0*_c^2)$
atomicmassunit	<code>_u</code>	$N(1.66053904e-27, 2e-35) * \_kg$
electronmass	<code>_m_e</code>	$N(9.10938356e-31, 1.1e-38) * \_kg$
protonmass	<code>_m_p</code>	$N(1.672621898e-27, 2.1e-35) * \_kg$
neutronmass	<code>_m_n</code>	$N(1.674927471e-27, 2.1e-35) * \_kg$
bohrmagneton	<code>_u_B</code>	$\_e*_h\_Pbar/(2*_m\_e)$
nuclearmagneton	<code>_u_N</code>	$\_e*_h\_Pbar/(2*_m\_p)$
electronmagneticmoment	<code>_u_e</code>	$N(-928.4764620e-26, 5.7e-32) * \_J/_T$
protonmagneticmoment	<code>_u_p</code>	$N(1.4106067873e-26, 9.7e-35) * \_J/_T$
neutronmagneticmoment	<code>_u_n</code>	$N(-0.96623650e-26, 2.3e-26) * \_J/_T$
finestructureconstant	<code>_alpha</code>	$\_u_0*_e^2*_c/(2*_h\_P)$
rydbergconstant	<code>_Ry</code>	$\_alpha^2*_m\_e*_c/(2*_h\_P)$
avogadronumber	<code>_N_A</code>	$6.02214076e23/_mol$
boltzmannconstant	<code>_k_B</code>	$1.380649e-23 * \_J/_K$
molargasconstant	<code>_R</code>	$N(8.3144598, 4.8e-6) * \_J/(\_K*_mol)$
stefanboltzmannconstant	<code>_sigma</code>	$\_Pi^2*_k\_B^4/(60*_h\_Pbar^3*_c^2)$
standardgravity	<code>_g_0</code>	$9.80665 * \_m/_s^2$

### 4.3 Coherent derived units in the SI

All units in this section are coherent derived units from the SI base units with special names, [2, 118].

Quantity	Unit	Symbol	Definition
Plane Angle <sup>1</sup>	radian	<code>_rad</code>	<code>_1</code>
Solid Angle <sup>2</sup>	steradian	<code>_sr</code>	<code>_rad^2</code>
Frequency	hertz	<code>_Hz</code>	<code>1/_s</code>
Force	newton	<code>_N</code>	<code>_kg*_m/_s^2</code>
Pressure	pascal	<code>_Pa</code>	<code>_N/_m^2</code>
Energy	joule	<code>_J</code>	<code>_N*_m</code>
Power	watt	<code>_W</code>	<code>_J/_s</code>
Electric Charge	coulomb	<code>_C</code>	<code>_A*_s</code>
Electric Potential	volt	<code>_V</code>	<code>_J/_C</code>
Electric Capacitance	farad	<code>_F</code>	<code>_C/_V</code>
Electric Resistance	ohm	<code>_Ohm</code>	<code>_V/_A</code>
Electric Conductance <sup>3</sup>	siemens	<code>_S</code>	<code>_A/_V</code>
Magnetic Flux	weber	<code>_Wb</code>	<code>_V*_s</code>
Magnetic Flux Density	tesla	<code>_T</code>	<code>_Wb/_m^2</code>
Inductance	henry	<code>_H</code>	<code>_Wb/_A</code>
Celsius Temperature <sup>4</sup>	celsius	<code>_degC</code>	<code>_K</code>
Luminous Flux	lumen	<code>_lm</code>	<code>_cd*_sr</code>
Illuminance	lux	<code>_lx</code>	<code>_lm/_m^2</code>
Activity	becquerel	<code>_Bq</code>	<code>1/_s</code>
Absorbed Dose	gray	<code>_Gy</code>	<code>_J/_kg</code>
Dose Equivalent	sievert	<code>_Sv</code>	<code>_J/_kg</code>
Catalytic Activity	katal	<code>_kat</code>	<code>_mol/_s</code>

<sup>1</sup>In the SI system, the quantity Plane Angle has the dimension of a number.

<sup>2</sup>In the SI system, the quantity Solid Angle has the dimension of a number.

<sup>3</sup>The unit `_PS` stands for peta siemens and is in conflict with the german version of the unit horsepower (Pferdestärke). Since the latter is more common than peta siemens, `_PS` is defined as the german version of horsepower.

<sup>4</sup>The unit `_degC` is by default interpreted as a temperature difference.

#### 4.4 Non-SI units accepted for use with the SI

There are a few units with dimension 1.

Quantity	Unit	Symbol	Definition
Time	minute	<code>_min</code>	$60 * \text{\_s}$
	hour	<code>_h</code>	$60 * \text{\_min}$
	day	<code>_d</code>	$24 * \text{\_h}$
Plane Angle	degree	<code>_deg</code>	$(\text{\_Pi}/180) * \text{\_rad}$
	arcminute	<code>_arcmin</code>	$\text{\_deg}/60$
	arcsecond	<code>_arcsec</code>	$\text{\_arcmin}/60$
Area	hectare	<code>_hectare</code>	$1e4 * \text{\_m}^2$
Volume	liter	<code>_L</code>	$1e-3 * \text{\_m}^3$
Mass	tonne	<code>_t</code>	$1e3 * \text{\_kg}$

#### 4.5 Nominal Astronomical Units

The nominal values of solar, terrestrial and jovial quantities are taken from IAU Resolution B3 [3].

Quantity	Unit	Symbol	Definition
Length	<code>nomsolradius</code>	<code>_R_S_nom</code>	$6.957e8 * \text{\_m}$
Irradiance	<code>nomsolirradiance</code>	<code>_S_S_nom</code>	$1361 * \text{\_W}/\text{\_m}^2$
Radiant Flux	<code>nomsolluminosity</code>	<code>_L_S_nom</code>	$3.828e26 * \text{\_W}$
Temperature	<code>nomsolefftemperature</code>	<code>_T_S_nom</code>	$5772 * \text{\_K}$
Mass Parameter	<code>nomsolmassparam</code>	<code>_GM_S_nom</code>	$1.3271244e20 * \text{\_m}^3 * \text{\_s}^{-2}$
Length	<code>nomterregradius</code>	<code>_Re_E_nom</code>	$6.3781e6 * \text{\_m}$
Length	<code>nomterrpolarradius</code>	<code>_Rp_E_nom</code>	$6.3568e6 * \text{\_m}$
Mass Parameter	<code>nomtermassparam</code>	<code>_GM_E_nom</code>	$3.986004e14 * \text{\_m}^3 * \text{\_s}^{-2}$
Length	<code>nomjovianeqradius</code>	<code>_Re_J_nom</code>	$7.1492e7 * \text{\_m}$
Length	<code>nomjovianpolarradius</code>	<code>_Rp_J_nom</code>	$6.6854e7 * \text{\_m}$
Mass Parameter	<code>nomjovianmassparam</code>	<code>_GM_J_nom</code>	$1.2668653e17 * \text{\_m}^3 * \text{\_s}^{-2}$

## 4.6 Other Non-SI units

The unit Bel is only available with prefix decibel, because `_B` is the unit byte.

Quantity	Unit	Symbol	Definition
Length	angstrom	<code>_angstrom</code>	$1e-10 * \text{\_m}$
	fermi	<code>_fermi</code>	$1e-15 * \text{\_m}$
	astronomicalunit	<code>_au</code>	$149597870700 * \text{\_m}$
	lightsecond	<code>_ls</code>	$\text{\_c} * \text{\_s}$
	lightyear	<code>_ly</code>	$\text{\_c} * \text{\_a}$
	parsec	<code>_pc</code>	$(648000/\text{\_Pi}) * \text{\_au}$
Area	barn	<code>_barn</code>	$1e-28 * \text{\_m}^2$
	are	<code>_are</code>	$1e2 * \text{\_m}^2$
Volume	metricteaspoon	<code>_tsp</code>	$5e-3 * \text{\_L}$
	metrictablespoon	<code>_Tbsp</code>	$3 * \text{\_tsp}$
Time	svedberg	<code>_svedberg</code>	$1e-13 * \text{\_s}$
	week	<code>_wk</code>	$7 * \text{\_d}$
	year	<code>_a</code>	$365.25 * \text{\_d}$
Plane Angle	gradian	<code>_gon</code>	$(\text{\_Pi}/200) * \text{\_rad}$
	turn	<code>_tr</code>	$2 * \text{\_Pi} * \text{\_rad}$
Solid Angle	spat	<code>_sp</code>	$4 * \text{\_Pi} * \text{\_sr}$
Force	kilopond	<code>_kp</code>	$\text{\_kg} * \text{\_g}_0$
Pressure	bar	<code>_bar</code>	$1e5 * \text{\_Pa}$
	standardatmosphere	<code>_atm</code>	$101325 * \text{\_Pa}$
	technicalatmosphere	<code>_at</code>	$\text{\_kp}/\text{\_cm}^2$
	millimeterofmercury	<code>_mmHg</code>	$133.322387415 * \text{\_Pa}$
	torr	<code>_Torr</code>	$(101325/760) * \text{\_Pa}$

Quantity	Unit	Symbol	Definition
Energy	thermochemicalcalorie	_cal	$4.184 * \_J$
	internationalcalorie	_cal_IT	$4.1868 * \_J$
	gramoftnt	_g_TNT	$1e3 * \_cal$
	tonoftnt	_t_TNT	$1e9 * \_cal$
	electronvolt	_eV	$\_e * \_V$
	wattsecond	_Ws	$\_W * \_s$
	watthour	_Wh	$\_W * \_h$
Power	voltampere	_VA	$\_V * \_A$
Electric Charge	amperesecond	_As	$\_A * \_s$
	amperehour	_Ah	$\_A * \_h$
Information	nibble	_nibble	$4 * \_bit$
	byte	_B	$8 * \_bit$
Information Transfer Rate	bitpersecond	_bps	$\_bit / \_s$
Number	percent	_percent	$1e-2 * \_1$
	permille	_permille	$1e-3 * \_1$
	partspermillion	_ppm	$1e-6 * \_1$
	partsperbillion	_ppb	$1e-9 * \_1$
	partspertrillion	_ppt	$1e-12 * \_1$
	partsperquadrillion	_ppq	$1e-15 * \_1$
	decibel	_dB	$\_1$
Power	metrichorsepower	_PS	$75 * \_g\_0 * \_kg * \_m / \_s$
Activity	curie	_Ci	$3.7e10 * \_Bq$
Absorbed Dose	rad	_Rad	$1e-2 * \_Gy$
Dose Equivalent	rem	_rem	$1e-2 * \_Sv$
Viscosity	poiseuille	_Pl	$\_Pa * \_s$

## 4.7 Imperial Units

Quantity	Unit	Symbol	Definition
Length	inch	<code>_in</code>	$2.54\text{e-}2 * \text{_m}$
	thou	<code>_th</code>	$1\text{e-}3 * \text{_in}$
DTP Point <sup>1</sup>	point	<code>_pt</code>	$\text{_in}/72$
	pica	<code>_pica</code>	$12 * \text{_pt}$
	hand	<code>_hh</code>	$4 * \text{_in}$
	foot	<code>_ft</code>	$12 * \text{_in}$
	yard	<code>_yd</code>	$3 * \text{_ft}$
	rod	<code>_rd</code>	$5.5 * \text{_yd}$
	chain	<code>_ch</code>	$4 * \text{_rd}$
	furlong	<code>_fur</code>	$10 * \text{_ch}$
	mile	<code>_mi</code>	$8 * \text{_fur}$
	league	<code>_lea</code>	$3 * \text{_mi}$
	nauticalmile	<code>_nmi</code>	$1852 * \text{_m}$
	nauticalleague	<code>_nlea</code>	$3 * \text{_nmi}$
	cable	<code>_cbl</code>	$0.1 * \text{_nmi}$
	fathom	<code>_ftm</code>	$6 * \text{_ft}$
Velocity	knot	<code>_kn</code>	$\text{_nmi}/\text{_h}$
Area	acre	<code>_ac</code>	$10 * \text{_ch}^2$
Volume	gallon	<code>_gal</code>	$4.54609 * \text{_L}$
	quart	<code>_qt</code>	$\text{_gal}/4$
	pint	<code>_pint</code>	$\text{_qt}/2$
	cup	<code>_cup</code>	$\text{_pint}/2$
	gill	<code>_gi</code>	$\text{_pint}/4$
	fluidounce	<code>_fl_oz</code>	$\text{_gi}/5$
	fluid dram	<code>_fl_dr</code>	$\text{_fl_oz}/8$

<sup>1</sup>The desktop publishing point or PostScript point is 1/72 of an international inch.

Quantity	Unit	Symbol	Definition
Mass	grain	<code>_gr</code>	$64.79891 \times \text{mg}$
	pound	<code>_lb</code>	$7000 \times \text{gr}$
	ounce	<code>_oz</code>	$\text{lb}/16$
	dram	<code>_dr</code>	$\text{lb}/256$
	stone	<code>_st</code>	$14 \times \text{lb}$
	quarter	<code>_qtr</code>	$2 \times \text{st}$
	hundredweight	<code>_cwt</code>	$4 \times \text{qtr}$
	longton	<code>_ton</code>	$20 \times \text{cwt}$



## 4.8 U.S. customary units

In the U.S., the length units are bound to the meter differently than in the imperial system. The following definitions are taken from [https://en.wikipedia.org/wiki/United\\_States\\_customary\\_units](https://en.wikipedia.org/wiki/United_States_customary_units).

Quantity	Unit	Symbol	Definition
Length	ussurveyinch	_in_US	_m/39.37
	ussurveyhand	_hh_US	4 * _in_US
	ussurveyfoot	_ft_US	3 * _hh_US
	ussurveylink	_li_US	0.66 * _ft_US
	ussurveyyard	_yd_US	3 * _ft_US
	ussurveyrod	_rd_US	5.5 * _yd_US
	ussurveychain	_ch_US	4 * _rd_US
	ussurveyfurlong	_fur_US	10 * _ch_US
	ussurveymile	_mi_US	8 * _fur_US
	ussurveyleague	_lea_US	3 * _mi_US
	ussurveyfathom	_ftm_US	72 * _in_US
	ussurveycable	_cbl_US	120 * _ftm_US
Area	ussurveyacre	_ac_US	_ch_US * _fur_US
Volume	usgallon	_gal_US	231 * _in^3
	usquart	_qt_US	_gal_US/4
	uspint	_pint_US	_qt_US/2
	uscup	_cup_US	_pint_US/2
	usgill	_gi_US	_pint_US/4
	usfluidounce	_fl_oz_US	_gi_US/4
	ustablespoon	_Tbsp_US	_fl_oz_US/2
	usteaspoon	_tsp_US	_Tbsp_US/3
	usfluid dram	_fl_dr_US	_fl_oz_US/8
Mass	usquarter	_qtr_US	25 * _lb
	ushundredweight	_cwt_US	4 * _qtr_US
	uston	_ton_US	20 * _cwt_US

## 4.9 International Currencies

International currency units based on exchange rates from 7.3.2019, 21:00 UTC.

Name	Symbol	Definition
AfghanAfghani	_AFN	$0.012 * \text{\_EUR}$
AfghanPul	_cAFN	$0.01 * \text{\_AFN}$
AlbanianLek	_ALL	$0.008 * \text{\_EUR}$
ArmenianDram	_AMD	$0.0018 * \text{\_EUR}$
ArmenianLuma	_cAMD	$0.01 * \text{\_AMD}$
AngolanKwanza	_AOA	$0.0028 * \text{\_EUR}$
AngolanCentimo	_cAOA	$0.01 * \text{\_AOA}$
ArgentinePeso	_ARS	$0.021 * \text{\_EUR}$
ArgentineCentavo	_cARS	$0.01 * \text{\_ARS}$
AustralianDollar	_AUD	$0.63 * \text{\_EUR}$
AustralianCent	_cAUD	$0.01 * \text{\_AUD}$
AzerbaijaniManat	_AZN	$0.63 * \text{\_EUR}$
AzerbaijaniQepik	_cAZN	$0.01 * \text{\_AZN}$
BosnianMark	_BAM	$0.51 * \text{\_EUR}$
BosnianFenings	_cBAM	$0.01 * \text{\_BAM}$
BangladeshiTaka	_BDT	$0.011 * \text{\_EUR}$
BangladeshiPoisha	_cBDT	$0.01 * \text{\_BDT}$
BurundianFranc	_BIF	$0.00049 * \text{\_EUR}$
BurundianCentime	_cBIF	$0.01 * \text{\_BIF}$
BolivianBoliviano	_BOB	$0.13 * \text{\_EUR}$
BolivianCentavo	_cBOB	$0.01 * \text{\_BOB}$
BrazilianReal	_BRL	$0.23 * \text{\_EUR}$
BrazilianCentavo	_cBRL	$0.01 * \text{\_BRL}$
BotswanaPula	_BWP	$0.083 * \text{\_EUR}$
BotswanaThebe	_cBWP	$0.01 * \text{\_BWP}$
BelarusianRuble	_BYN	$0.42 * \text{\_EUR}$
BelarusianKapiejka	_cBYN	$0.01 * \text{\_BYN}$

Name	Symbol	Definition
CanadianDollar	_CAD	0.66 * _EUR
CanadianCent	_cCAD	0.01 * _CAD
CongoleseFranc	_CDF	0.00055 * _EUR
CongoleseCentime	_cCDF	0.01 * _CDF
SwissFranc	_CHF	0.88 * _EUR
SwissRappen	_cCHF	0.01 * _CHF
ChileanPeso	_CLP	0.0013 * _EUR
ChileanCentavo	_cCLP	0.01 * _CLP
ChineseRenminbiYuan	_CNY	0.13 * _EUR
ChineseRenminbiFen	_cCNY	0.01 * _CNY
ColombianPeso	_COP	0.00028 * _EUR
ColombianCentavo	_cCOP	0.01 * _COP
CostaRicanColon	_CRC	0.0015 * _EUR
CostaRicanCentimos	_cCRC	0.01 * _CRC
CzechKoruna	_CZK	0.039 * _EUR
CzechHaler	_cCZK	0.01 * _CZK
DanishKrone	_DKK	0.13 * _EUR
DanishOre	_cDKK	0.01 * _DKK
DominicanPeso	_DOP	0.018 * _EUR
DominicanCentavo	_cDOP	0.01 * _DOP
AlgerianDinar	_DZD	0.0074 * _EUR
AlgerianSanteem	_cDZD	0.01 * _DZD
EgyptianPound	_EGP	0.051 * _EUR
EgyptianPiastre	_cEGP	0.01 * _EGP
EthiopianBirr	_ETB	0.031 * _EUR
EthiopianSantim	_cETB	0.01 * _ETB
FijianDollar	_FJD	0.42 * _EUR
FijianCent	_cFJD	0.01 * _FJD

Name	Symbol	Definition
PoundSterling	_GBP	1.16 * _EUR
PennySterling	_cGBP	0.01 * _GBP
GeorgianLari	_GEL	0.33 * _EUR
GeorgianTetri	_cGEL	0.01 * _GEL
GhanaianCedi	_GHS	0.16 * _EUR
GhanaianPesewa	_cGHS	0.01 * _GHS
GambianDalasi	_GMD	0.018 * _EUR
GambianButut	_cGMD	0.01 * _GMD
GuineanFranc	_GNF	9.6e-05 * _EUR
GuineanCentime	_cGNF	0.01 * _GNF
GuatemalanQuetzal	_GTQ	0.12 * _EUR
GuatemalanCentavo	_cGTQ	0.01 * _GTQ
GuyaneseDollar	_GYD	0.0043 * _EUR
GuyaneseCent	_cGYD	0.01 * _GYD
HongKongDollar	_HKD	0.11 * _EUR
HongKongCent	_cHKD	0.01 * _HKD
HonduranLempira	_HNL	0.036 * _EUR
HonduranCentavo	_cHNL	0.01 * _HNL
CroatianKuna	_HRK	0.13 * _EUR
CroatianLipa	_cHRK	0.01 * _HRK
HaitianGourde	_HTG	0.011 * _EUR
HaitianCentime	_cHTG	0.01 * _HTG
HungarianForint	_HUF	0.0032 * _EUR
HungarianFiller	_cHUF	0.01 * _HUF
IndonesianRupiah	_IDR	6.2e-05 * _EUR
IndonesianSen	_cIDR	0.01 * _IDR
IsraeliNewShekel	_ILS	0.25 * _EUR
IsraeliNewAgora	_cILS	0.01 * _ILS

Name	Symbol	Definition
IndianRupee	_INR	0.013 * _EUR
IndianPaissa	_cINR	0.01 * _INR
IraqiDinar	_IQD	0.00074 * _EUR
IraqiFils	_cIQD	0.001 * _IQD
IranianRial	_IRR	2.7e-05 * _EUR
IranianToman	_cIRR	10.0 * _IRR
IcelandicKrona	_ISK	0.0073 * _EUR
JamaicanDollar	_JMD	0.007 * _EUR
JamaicanCent	_cJMD	0.01 * _JMD
JapaneseYen	_JPY	0.008 * _EUR
KenyanShilling	_KES	0.0089 * _EUR
KenyanCent	_cKES	0.01 * _KES
KyrgyzstaniSom	_KGS	0.013 * _EUR
KyrgyzstaniTyin	_cKGS	0.01 * _KGS
CambodianRiel	_KHR	0.00022 * _EUR
NorthKoreanWon	_KPW	0.00099 * _EUR
NorthKoreanChon	_cKPW	0.01 * _KPW
SouthKoreanWon	_KRW	0.00078 * _EUR
SouthKoreanJeon	_cKRW	0.01 * _KRW
KuwaitiDinar	_KWD	2.93 * _EUR
KuwaitiFils	_cKWD	0.001 * _KWD
KazakhstaniTenge	_KZT	0.0023 * _EUR
KazakhstaniTiyn	_cKZT	0.01 * _KZT
LaoKip	_LAK	0.0001 * _EUR
LaoAtt	_cLAK	0.01 * _LAK
SriLankanRupee	_LKR	0.005 * _EUR
SriLankanCent	_cLKR	0.01 * _LKR
LiberianDollar	_LRD	0.0055 * _EUR
LiberianCent	_cLRD	0.01 * _LRD

Name	Symbol	Definition
LibyanDinar	_LYD	0.64 * _EUR
LibyanDirham	_cLYD	0.001 * _LYD
MoroccanDirham	_MAD	0.092 * _EUR
MoroccanSantim	_cMAD	0.01 * _MAD
MoldovanLeu	_MDL	0.052 * _EUR
MoldovanBan	_cMDL	0.01 * _MDL
MalagasyAriary	_MGA	0.00025 * _EUR
MalagasyIraimbilanja	_cMGA	0.2 * _MGA
MacedonianDenar	_MKD	0.016 * _EUR
MacedonianDeni	_cMKD	0.01 * _MKD
BurmeseKyat	_MMK	0.00059 * _EUR
BurmesePya	_cMMK	0.01 * _MMK
MongolianTogrog	_MNT	0.00034 * _EUR
MongolianMongol	_cMNT	0.01 * _MNT
MauritanianOuguiya	_MRU	0.025 * _EUR
MauritanianKhoudms	_cMRU	0.2 * _MRU
MauritianRupee	_MUR	0.025 * _EUR
MauritianCent	_cMUR	0.01 * _MUR
MaldivianRufiyaa	_MVR	0.058 * _EUR
MaldivianLaari	_cMVR	0.01 * _MVR
MalawianKwacha	_MWK	0.0012 * _EUR
MalawianTambala	_cMWK	0.01 * _MWK
MexicanPeso	_MXN	0.046 * _EUR
MexicanCentavo	_cMXN	0.01 * _MXN
MalaysianRinggit	_MYR	0.22 * _EUR
MalaysianSen	_cMYR	0.01 * _MYR
MozambicanMetical	_MZN	0.014 * _EUR
MozambicanCentavo	_cMZN	0.01 * _MZN

Name	Symbol	Definition
NigerianNaira	_NGN	0.0025 * _EUR
NigerianKobo	_cNGN	0.01 * _NGN
NicaraguanCordoba	_NIO	0.027 * _EUR
NicaraguanCentavo	_cNIO	0.01 * _NIO
NorwegianKrone	_NOK	0.1 * _EUR
NorwegianOre	_cNOK	0.01 * _NOK
NewZealandDollar	_NZD	0.61 * _EUR
NewZealandCent	_cNZD	0.01 * _NZD
PeruvianSol	_PEN	0.27 * _EUR
PeruvianCentimo	_cPEN	0.01 * _PEN
PapuaNewGuineanKina	_PGK	0.26 * _EUR
PapuaNewGuineanToea	_cPGK	0.01 * _PGK
PhilippinePeso	_PHP	0.017 * _EUR
PhilippineSentimo	_cPHP	0.01 * _PHP
PakistaniRupee	_PKR	0.0064 * _EUR
PakistaniPaissa	_cPKR	0.01 * _PKR
PolishZloty	_PLN	0.23 * _EUR
PolishGrosz	_cPLN	0.01 * _PLN
ParaguayanGuarani	_PYG	0.00015 * _EUR
ParaguayanCentimo	_cPYG	0.01 * _PYG
QatariRiyal	_QAR	0.24 * _EUR
QatariDirham	_cQAR	0.01 * _QAR
RomanianLeu	_RON	0.21 * _EUR
RomanianBan	_cRON	0.01 * _RON
SerbianDinar	_RSD	0.0085 * _EUR
SerbianPara	_cRSD	0.01 * _RSD
RussianRuble	_RUB	0.013 * _EUR
RussianKopeyka	_cRUB	0.01 * _RUB

Name	Symbol	Definition
RwandanFranc	_RWF	0.00098 * _EUR
RwandanCentime	_cRWF	0.01 * _RWF
SolomonIslandsDollar	_SBD	0.11 * _EUR
SolomonIslandsCent	_cSBD	0.01 * _SBD
SeychelloisRuppee	_SCR	0.065 * _EUR
SeychelloisCent	_cSCR	0.01 * _SCR
SudanesePound	_SDG	0.019 * _EUR
SudaneseQirsh	_cSDG	0.01 * _SDG
SwedishKrona	_SEK	0.094 * _EUR
SwedishOre	_cSEK	0.01 * _SEK
SingaporeDollar	_SGD	0.65 * _EUR
SingaporeCent	_cSGD	0.01 * _SGD
SierraLeoneanLeone	_SLL	0.0001 * _EUR
SierraLeoneanCent	_cSLL	0.01 * _SLL
SomalilandShilling	_SQS	0.00013 * _EUR
SomalilandCent	_cSQS	0.01 * _SQS
SomaliShilling	_SOS	0.0015 * _EUR
SomaliSenti	_cSOS	0.01 * _SOS
SurinameseDollar	_SRD	0.12 * _EUR
SurinameseCent	_cSRD	0.01 * _SRD
SyrianPound	_SYP	0.0017 * _EUR
SyrianPiastre	_cSYP	0.01 * _SYP
ThaiBaht	_THB	0.028 * _EUR
ThaiSatang	_cTHB	0.01 * _THB
TajikistaniSamani	_TJS	0.094 * _EUR
TajikistaniDiram	_cTJS	0.01 * _TJS
Tonganpaanga	_TOP	0.397 * _EUR
TonganSeniti	_cTOP	0.01 * _TOP



Name	Symbol	Definition
TurkishLira	_TRY	0.16 * _EUR
TurkishKurus	_cTRY	0.01 * _TRY
TrinidadAndTobagoDollar	_TTD	0.13 * _EUR
TrinidadAndTobagoCent	_cTTD	0.01 * _TTD
NewTaiwanDollar	_TWD	0.029 * _EUR
NewTaiwanCent	_cTWD	0.01 * _TWD
TanzanianShilling	_TZS	0.00038 * _EUR
TanzanianSenti	_cTZS	0.01 * _TZS
UkrainianHryvnia	_UAH	0.00038 * _EUR
UkrainianKopiyka	_cUAH	0.01 * _UAH
UgandanShilling	_UGX	0.00024 * _EUR
UgandanCent	_cUGX	0.01 * _UGX
USDollar	_USD	0.89 * _EUR
USCent	_cUSD	0.01 * _USD
UruguayanPeso	_UYU	0.027 * _EUR
UruguayanCentesimo	_cUYU	0.01 * _UYU
UzbekistaniSom	_UZS	0.00011 * _EUR
UzbekistaniTiyin	_cUZS	0.01 * _UZS
VenezuelanBolivarSoberano	_VES	0.0003 * _EUR
VenezuelanCentimoSoberano	_cVES	0.01 * _VES
VietnameseDong	_VND	3.8e-05 * _EUR
VietnameseXu	_cVND	0.01 * _VND
SamoanTala	_WST	0.34 * _EUR
SamoanSene	_cWST	0.01 * _WST
YemeniRial	_YER	0.0036 * _EUR
YemeniDinar	_cYER	0.01 * _YER
SouthAfricanRand	_ZAR	0.062 * _EUR
SouthAfricanCent	_cZAR	0.01 * _ZAR
ZambianKwacha	_ZMW	0.074 * _EUR
ZambianNgwee	_cZMW	0.01 * _ZMW

#### 4.9.1 Pegged International Currencies

International currency which are pegged to other currencies.

Name	Symbol	Definition
UnitedArabEmiratesDirham	_AED	$(1/3.6725) * \_USD$
UnitedArabEmiratesFils	_cAED	$0.01 * \_AED$
NetherlandsAntilleanGuilder	_ANG	$(1/1.79) * \_USD$
NetherlandsAntilleanCent	_cANG	$0.01 * \_ANG$
ArubanFlorin	_AWG	$(1/1.79) * \_USD$
ArubanCent	_cAWG	$0.01 * \_AWG$
BarbadianDollar	_BBD	$0.5 * \_USD$
BarbadianCent	_cBBD	$0.01 * \_BBD$
BulgarianLev	_BGN	$0.51129 * \_EUR$
BulgarianStotinka	_cBGN	$0.01 * \_BGN$
BahrainiDinar	_BHD	$(1/0.376) * \_USD$
BahrainiFils	_cBHD	$0.001 * \_BHD$
BermudianDollar	_BMD	$1 * \_USD$
BermudianCent	_cBMD	$0.01 * \_BMD$
BruneiDollar	_BND	$1 * \_SGD$
BruneiSen	_cBND	$0.01 * \_BND$
BahamianDollar	_BSD	$1 * \_USD$
BahamianCent	_cBSD	$0.01 * \_BSD$
BhutaneseNgultrum	_BTN	$1 * \_INR$
BhutaneseChhertum	_cBTN	$0.01 * \_BTN$
BelizeDollar	_BZD	$0.5 * \_USD$
BelizeCent	_cBZD	$0.01 * \_BZD$
CubanoConvertiblePeso	_CUC	$1 * \_USD$
CubanoConvertibleCentavo	_cCUC	$0.01 * \_CUC$
CubanPeso	_CUP	$(1/24) * \_CUC$
CubanCentavo	_cCUP	$0.01 * \_CUP$

Name	Symbol	Definition
CapeVerdeanEscudo	_CVE	$(1/110.265) * \_EUR$
CapeVerdeanCentavo	_cCVE	$0.01 * \_CVE$
DjiboutianFranc	_DJF	$(1/177.721) * \_USD$
DjiboutianCentime	_cDJF	$0.01 * \_DJF$
EritreanNakfa	_ERN	$(1/15) * \_USD$
EritreanCent	_cERN	$0.01 * \_ERN$
FalklandIslandsPound	_FKP	$1 * \_GBP$
FalklandIslandsPenny	_cFKP	$0.01 * \_FKP$
GuernseyPound	_GGP	$1 * \_GBP$
GuernseyPenny	_cGGP	$0.01 * \_GGP$
GibraltarPound	_GIP	$1 * \_GBP$
GibraltarPenny	_cGIP	$0.01 * \_GIP$
ManxPound	_IMP	$1 * \_GBP$
ManxPenny	_cIMP	$0.01 * \_IMP$
JerseyPound	_JEP	$1 * \_GBP$
JerseyPenny	_cJEP	$0.01 * \_JEP$
JordanianDinar	_JOD	$(1/0.708) * \_USD$
JordanianFils	_cJOD	$0.001 * \_JOD$
KiribatiDollar	_KID	$1 * \_AUD$
KiribatiCent	_cKID	$0.01 * \_KID$
Comorianfranc	_KMF	$(1/491.96775) * \_EUR$
ComorianCentime	_cKMF	$0.01 * \_KMF$
CaymanIslandsDollar	_KYD	$1.2 * \_USD$
CaymanIslandsCent	_cKYD	$0.01 * \_KYD$
LebanesePound	_LBP	$(1/1507.5) * \_USD$
LebaneseQeresh	_cLBP	$0.01 * \_LBP$
MacanesePataca	_MOP	$(1/1.03) * \_HKD$
MacaneseAvo	_cMOP	$0.01 * \_MOP$

Name	Symbol	Definition
NamibianDollar	_NAD	1 * _ZAR
NamibianCent	_cNAD	0.01 * _NAD
NepaleseRupee	_NPR	(1/1.6) * _INR
NepalesePaissa	_cNPR	0.01 * _NPR
OmaniRial	_OMR	(1/2.6008) * _USD
OmaniBaisa	_cOMR	0.001 * _OMR
PanamanianBalboa	_PAB	1 * _USD
PanamanianCentesimo	_cPAB	0.01 * _PAB
TransnistrianRuble	_PRB	(1/16.1) * _USD
TransnistrianKopeck	_cPRB	0.01 * _PRB
SaudiRiyal	_SAR	(1/3.75) * _USD
SaudiHalalah	_cSAR	0.01 * _SAR
SaintHelenaPound	_SHP	1 * _GBP
SaintHelenaPenny	_cSHP	0.01 * _SHP
SouthSudanesePound	_SSP	1 * _SDG
SouthSudanesePiaster	_cSSP	0.01 * _SSP
SaoTomeAndPrincipeDobra	_STN	(1/24.5) * _EUR
SaoTomeAndPrincipeCentimo	_cSTN	0.01 * _STN
SwaziLilangeni	_SZL	1 * _ZAR
SwaziCent	_cSZL	0.01 * _SZL
TurkmenistanManat	_TMT	(1/3.5) * _USD
TurkmenistanTenge	_cTMT	0.01 * _TMT
TuvaluanDollar	_TVD	1 * _AUD
TuvaluanCent	_cTVD	0.01 * _TVD
CentralAfricanCFAFranc	_XAF	(1/655.957) * _EUR
CentralAfricanCFACentime	_cXAF	0.01 * _XAF
EasternCaribbeanDollar	_XCD	(1/2.7) * _USD
EasternCaribbeanCent	_cXCD	0.01 * _XCD

Name	Symbol	Definition
WestAfricanCFAFranc	_XOF	$(1/655.957) * \text{\_USD}$
WestAfricanCFACentime	_cXOF	$0.01 * \text{\_XOF}$
CFPFranc	_XPF	$(1000/8.38) * \text{\_EUR}$
CFPCentime	_cXPF	$0.01 * \text{\_XPF}$
ZimbabweanBonds	_ZWL	$1 * \text{\_USD}$
ZimbabweanCent	_cZWL	$0.01 * \text{\_ZWL}$

## 5 Lua Documentation

In this chapter, the following shortcuts will be used.

```
1 local D = physical.Dimension
2 local U = physical.Unit
3 local N = physical.Number
4 local Q = physical.Quantity
```

The term **number** refers to a lua integer or a lua float number. By **string** a lua string is meant and by **bool** a lua boolean.

### 5.1 physical.Quantity

The quantity class is the main part of the library. Each physical Quantity and all units are represented by an instance of this class.

#### **Q.new(q=nil)**

Copy Constructor

**q** : Q or **number**, optional  
Optional argument is either Q, a **number** or **nil**.  
**return** : Q  
The created Q instance

As an argument it takes Q, **number** or **nil**. If Q is given, a copy of it is made and returned. If a **number** is given, the function creates a dimensionless quantity with that value. In the case **nil** is given, the quantity **\_1** is returned.

#### **Example**

```
1 myOne = Q()
2 myNumber = Q(42)
3 myLength = Q(73*_m)
```

#### **Q.defineBase(symbol,name,dimension)**

This function is used to declare base quantities from which all other quantities are derived from.

**symbol** : string  
The symbol of the base quantity.  
**name** : string  
The name of the base quantity.  
**dimension** : D  
An instance of the D class, which represents the dimension of the quantity.

```
return : Q
    The created Q instance.
```

The function creates a global variable of the created base quantity. The name consist of an underscore concatenated with the **symbol** argument, i.e. the symbol **m** becomes the global variable **\_m**.

The **name** is used for example in the siunitx conversion function, e.g **meter** will be converted to **\meter**.

Each quantity has a dimension associated with it. The argument **dimension** allows any dimension to be associated to base quantities.

### Example

```
1 Q.defineBase("m", "meter", L)
2 Q.defineBase("kg", "kilogram", M)
```

## Quantity.define(symbol, name, q)

Creates a new derived quantity from an expression of other quantities. Affine quantities like the absolute temperature in celsius are not supported.

```
symbol : string
    Symbol of the base quantity
name : string
    The Name of the derived quantity.
q : physical.Quantity
    The definition of the derived quantity.
return : Quantity
    The created quantity.
```

The function creates a global variable of the created base quantity. The name consist of an underscore concatenated with the **symbol** argument, i.e. the symbol **N** becomes the global variable **\_N**.

The **name** is used for example in the siunitx conversion function, e.g **newton** will be converted to **\newton**.

### Example

```
1 Q.define("L", "liter", _dm^3)
2 Q.define("Pa", "pascal", _N/_m^2)
3 Q.define("C", "coulomb", _A*_s)
4
5 Q.define("degC", "celsius", _K)
```

### **Quantity.defineAlias(symbol, q)**

Creates an alias of a quantity.

**symbol** : string  
The name of the alias.

**q** : Quantity  
The aliased quantity.

```
1 Q.defineAlias("degC_0", 273.15*_degC)
```

### **Quantity.definePrefix(symbol,name,factor)**

Defines a new prefix.

**symbol** : string  
Symbol of the base quantity

**name** : string  
Name of the base quantity

**factor** : number  
The factor which corresponds to the prefix

```
1 Q.definePrefix("c", "centi", 1e-2)
2 Q.definePrefix("a", "atto", 1e-18)
```

### **Quantity.addPrefix(prefixes, units)**

Create several units with prefixes from a given unit.

**prefixes** : string  
A list of unit symbols.

**units** : Quantity  
A list of quantities.

```
1 Q.addPrefix({"n","u","m","k","M","G"},{_m,_s,_A})
```

### **Quantity.to(self,q)**

Converts the quantity self to the unit of the quantity q.

**self** : Quantity  
A quantity.

**q** : Quantity The quantity to wich **self** will be converted to.



```

1 s = 1.9 * _km
2 print( s:to(_m) )
3 1900.0 * _m
4
5 T = 10 * _degC
6 print( T:to(_K) )
7 10.0 * _K
8 print( T:to(_K) )
9 10.0 * _K

```

### **Quantity.tosiunitx(self,param,mode)**

Converts the quantity into a siunitx string.

```

self : Quantity
param : string
mode : Number, 0:\SI, 1:\num, 2:\si

```

```

1 s = 1.9 * _km
2
3 print( s:tosiunitx() )
4 \SI{1.9}{\kilo\meter}
5
6 print( s:tosiunitx(nil,1) )
7 \num{1.9}
8
9 print( s:tosiunitx(nil,2) )
10 \si{\kilo\meter}

```

### **Quantity.isclose(self,q,r)**

Checks if this quantity is close to another one. The argument **r** is the maximal relative deviation.

```

self : Quantity
q : Quantity, Number
r : Number

```

```

1 s_1 = 1.9 * _m
2 s_2 = 2.0 * _m
3 print( s_1:isclose(s_2,0.1) )
4 true
5 print( s_1:isclose(s_2,0.01) )
6 false

```

### **Quantity.min(q1, q2, ...)**

Returns the smallest quantity of several given ones. The function returns **q1** if the Quantities are equal.

**q1** : Quantity, Number, first argument

**q2** : Quantity, Number, second argument

```
1 s_1 = 15 * _m
2 s_2 = 5 * _m
3 print(s_1:min(s_2))
4 5 * _m
```

### **Quantity.max(q1, q2, ...)**

Returns the biggest quantity of several given ones. The function returns **q1** if the Quantities are equal.

**q1** : Quantity, Number, first argument

**q2** : Quantity, Number, second argument

```
1 s_1 = 15 * _m
2 s_2 = 5 * _m
3 print(s_1:max(s_2))
4 15 * _m
```

### **Quantity.abs(q)**

Returns the absolute value of the given quantity **q**.

**q** : Quantity, Number, argument

```
1 U = -5 * _V
2 print(U)
3 -5 * _V
4 print(U:abs())
5 5 * _V
```

### **Quantity.sqrt(q)**

Returns the square root of the given quantity.

**q** : Quantity, Number argument

```
1 A = 25 * _m^2
2 s = sqrt(A)
3 print(s)
4
```

### **Quantity.log(q, base)**

Returns the logarithm of the given quantity. If no base is given, the natural logarithm is calculated.

**q** : Quantity, Number dimensionless argument  
**base** : Quantity, Number dimensionless argument

```
1 I = 1 * _W/_m^2
2 I_0 = 1e-12 * _W/_m^2
3 print(10 * (I/I_0):log(10) * _dB )
4 120.0 * _dB
```

### **Quantity.exp(q)**

Returns the value of the exponential function of the given quantity.

**q** : Quantity, Number dimensionless argument

```
1 x = 2 * _1
2 print( x:exp() )
3 7.3890560989307
```

### **Quantity.sin(q)**

Returns the value of the sinus function of the given quantity.

**q** : Quantity, Number dimensionless argument

```
1 alpha = 30 * _deg
2 print( alpha:sin() )
3 0.5
```

### **Quantity.cos(q)**

Returns the value of the cosinus function of the given quantity. The quantity has to be dimensionless.

**q** : Quantity, Number dimensionless argument

```
1 alpha = 60 * _deg
2 print( alpha:cos() )
3 0.5
```

### **Quantity.tan(q)**

Returns the value of the tangent function of the given quantity. The quantity has to be dimensionless.

`q : Quantity, Number dimensionless argument`

```
1 alpha = 45 * _deg
2 print( alpha:tan() )
3 1.0
```

### **Quantity.asin(q)**

Returns the value of the arcus sinus function of the given quantity. The quantity has to be dimensionless.

`q : Quantity, Number dimensionless argument`

```
1 x = 0.5 * _1
2 print( x:asin():to(_deg) )
3 30.0 * _deg
```

### **Quantity.acos(q)**

Returns the value of the arcus cosinus function of the given quantity. The quantity has to be dimensionless.

`q : Quantity, Number dimensionless argument`

```
1 x = 0.5 * _1
2 print( x:acos():to(_deg) )
3 60.0 * _deg
```

### **Quantity.atan(q)**

Returns the value of the arcus tangent function of the given quantity. The quantity has to be dimensionless.

`q : Quantity, Number dimensionless argument`

```
1 x = 1 * _1
2 print( x:atan():to(_deg) )
3 45.0 * _deg
```

### **Quantity.sinh(q)**

Returns the value of the hyperbolic sine function of the given quantity. The quantity has to be dimensionless. Since lua doesn't implement the hyperbolic functions the following formula is used

$$\sinh(x) = 0.5 \cdot e^x - 0.5/e^x \quad .$$

**q** : **Quantity**, **Number** dimensionless argument

```
1  x = 1 * _1
2  print( x:sinh() )
3  1.1752011936438
```

### **Quantity.cosh(q)**

Returns the value of the hyperbolic cosine function of the given quantity. The quantity has to be dimensionless. Since lua doesn't implement the hyperbolic functions the following formula is used

$$\cosh(x) = 0.5 \cdot e^x + 0.5/e^x \quad .$$

**q** : **Quantity**, **Number** dimensionless argument

```
1  x = 1 * _1
2  print( x:cosh() )
3  1.5430806348152
```

### **Quantity.tanh(q)**

Returns the value of the hyperbolic tangent function of the given quantity. The quantity has to be dimensionless. Since lua doesn't implement the hyperbolic functions the following formula is used

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad .$$

**q** : **Quantity**, **Number** dimensionless argument

```
1  x = 1 * _1
2  print( x:tanh() )
3  0.76159415595576
```

### **Quantity.asinh(q)**

Returns the value of the inverse hyperbolic sine function of the given quantity. The quantity has to be dimensionless. Since lua doesn't implement the hyperbolic functions the following formula is used

$$\operatorname{asinh}(x) = \ln \left( x + \sqrt{x^2 + 1} \right) \quad .$$

`q : Quantity, Number` dimensionless argument

```
1 x = 1 * _1
2 print( x:asinh() )
3 0.88137358701954
```

### **Quantity.acosh(q)**

Returns the value of the inverse hyperbolic cosine function of the given quantity. The quantity has to be dimensionless. Since lua doesn't implement the hyperbolic functions the following formula is used

$$\operatorname{acosh}(x) = \ln \left( x + \sqrt{x^2 - 1} \right) \quad , x > 1 \quad .$$

`q : Quantity, Number` dimensionless argument bigger than or equal to one.

```
1 x = 2 * _1
2 print( x:acosh() )
3 1.3169578969248
```

### **Quantity.atanh(q)**

Returns the value of the inverse hyperbolic cosine function of the given quantity. The quantity has to be dimensionless. Since lua doesn't implement the hyperbolic functions the following formula is used

$$\operatorname{atanh}(x) = \ln \left( \frac{1+x}{1-x} \right) \quad , -1 < x < 1 \quad .$$

`q : Quantity, Number` dimensionless argument with magnitude smaller than one.

```
1 x = 0.5 * _1
2 print( x:atanh() )
3 0.54930614433405
```

## **5.2 physical.Dimension**

All physical quantities do have a physical dimension. For example the quantity *Area* has the dimension  $L^2$  (length to the power of two). In the SI-System there are seven base dimensions, from which all other dimensions are derived. Each dimension is represented by an  $n$ -tuple, where  $n$  is the number of base dimensions. Each physical quantity has an associated dimension object. It is used to check equality and if addition or subtraction is allowed.

### **Dimension.new(q=nil)**

Constructor of the **Dimension** class.

**q** : **Dimension** or **string**, optional  
The name or symbol of the dimension. If **q** is a dimension, a copy of it is made. If no argument is given, a dimension *zero* is created.

**return** : **Dimension**  
The created **Quantity** object

#### **Example**

```
1  V_1 = D("Velocity")
2  L = D("L")
3  V_2 = D(L/T)
```

## **5.3 physical.Unit**

The task of this class is keeping track of the unit term. The unit term is a fraction of units. The units in the numerator and denominator can have an exponent.

### **Unit.new(u=nil)**

Copy Constructor. It copies a given unit object. If nothing is given, an empty unit is created.

**u** : **Unit**  
The unit object which will be copied.

**return** : **Unit**  
The created **Unit** object

### **Unit.new(symbol, name, prefixsymbol=nil, prefixname=nil)**

Constructor. A new **Unit** object with **symbol** is created. The **prefixsymbol** and **prefixname** are optional.

**symbol** : **String**  
The symbol of the unit.

**name** : **String**  
The name of the unit.

**prefixsymbol** : **String**  
The optional symbol of the prefix.

**prefixname** : **String**  
The optional name of the prefix.

**return** : **Unit**  
The created **Unit** object

### **Unit.tosiunitx(self)**

The unit term will be compiled into a string, which the LaTeX package siunitx can understand.

```
return : String
    The siunitx representation of the unit term.
```

## **5.4 physical.Number**

It does arithmetics with gaussian error propagation. A number instance has a mean value called **x** and an uncertainty called **dx**.

### **Number.new(n=nil)**

This is the copy Constructor. It copies a given number object. If **n** is **nil**, an instance representing number zero with uncertainty zero is created.

```
n : Number
    The number object to be copied.

return : Number
    The created Number instance.
```

### **Number.new(x, dx)**

This constructor, creates a new instance with mean value **x** and uncertainty **dx**.

```
x : number
    mean value

dx : number
    uncertainty value

return : Number
    The created Number instance.
```

### **Example**

```
1  n = N(12,0.1)
2  print(n)
```



### **Number.new(str)**

This constructor creates a new instance from a string. It can parse strings of the form 3.4, 3.4e-3, 5.4e-3 +/- 2.4e-6, 5.45(7)e-23.

#### **Parameters / Return**

**str** : string  
The number as a string.

**return** : Number  
The created Number object

#### **Example**

```
1  n_1 = N("12.3e-6")
2  print(n_1)
3
4  n_2 = N("12 +/- 0.1")
5  print(n_2)
6
7  n_3 = N("12.0(1)")
8  print(n_3)
```

### **Number.mean(n)**

Returns the mean value

#### **Parameters / Return**

**return** : number  
The mean value

### **Number.uncertainty(n)**

Returns the uncertainty value

#### **Parameters / Return**

**return** : number  
The uncertainty value

### **Number.abs(n)**

Returns the absolute value of the number.

#### **Parameters / Return**

**return** : number  
The absolute value

### **Number.sqrt(n)**

Returns the square root of the number.

#### **Parameters / Return**

**return** : number

The square root

## References

- [1] Webpage <https://physics.nist.gov/cuu/index.html>, August 2019.
- [2] Bureau International des Poids et Mesures. The international system of units (si), 2006.
- [3] Prša et al. Nominal values for selected solar and planetary quantities: Iau 2015 resolution b3. The Astronomical Journal, 152:41, August 2016.

## Index

acre `_ac`, 15  
amperehour `_Ah`, 14  
amperesecond `_As`, 14  
angstrom `_angstrom`, 13  
arcminute `_arcmin`, 12  
arcsecond `_arcsec`, 12  
are `_are`, 13  
astronomicalunit `_au`, 13  
atomicmassunit , 10  
avogadronumber , 10  
  
bar `_bar`, 13  
barn `_barn`, 13  
becquerel `_Bq`, 11  
bitpersecond `_bps`, 14  
bohrmagneton , 10  
boltzmannconstant , 10  
byte `_B`, 14  
  
cable `_cbl`, 15  
celsius `_degC`, 11  
chain `_ch`, 15  
coulomb `_C`, 11  
cup `_cup`, 15  
curie `_Ci`, 14  
  
day `_d`, 12  
decibel `_dB`, 14  
degree `_deg`, 12  
dram `_dr`, 16  
  
electronmagneticmoment , 10  
electronmass , 10  
electronvolt `_eV`, 14  
elementarycharge , 10  
eulersnumber , 10  
  
farad `_F`, 11  
fathom `_ftm`, 15  
fermi `_fermi`, 13  
finestructureconstant , 10  
fluidram `_fl_dr`, 15  
fluidounce `_fl_oz`, 15  
  
foot `_ft`, 15  
furlong `_fur`, 15  
  
gallon `_gal`, 15  
gill `_gi`, 15  
gradian `_gon`, 13  
grain `_gr`, 16  
gramoftnt `_g_TNT`, 14  
gravitationalconstant , 10  
gray `_Gy`, 11  
  
hand `_hh`, 15  
hectare `_hectare`, 12  
henry `_H`, 11  
hertz `_Hz`, 11  
hour `_h`, 12  
hundredweight `_cwt`, 16  
  
inch `_in`, 15  
internationalcalorie `_cal_IT`, 14  
  
joule `_J`, 11  
  
katal `_kat`, 11  
kilopond `_kp`, 13  
knot `_kn`, 15  
  
league `_lea`, 15  
lightsecond `_ls`, 13  
lightyear `_ly`, 13  
liter `_L`, 12  
longton `_ton`, 16  
lumen `_lm`, 11  
lux `_lx`, 11  
  
metrichorsepower `_PS`, 14  
metrictablespoon `_Tbsp`, 13  
metricteaspoon `_tsp`, 13  
mile `_mi`, 15  
millimeterofmercury `_mmHg`, 13  
minute `_min`, 12  
molargasconstant , 10  
  
nauticalleague `_nlea`, 15

nauticalmile `_nmi`, 15  
 neutronmagneticmoment , 10  
 neutronmass , 10  
 newton `_N`, 11  
 nibble `_nibble`, 14  
 nomjovianeqradius `_Re_J_nom`, 12  
 nomjovianmassparam `_GM_J_nom`, 12  
 nomjovianpolradius `_Rp_J_nom`, 12  
 nomsoleffttemperature `_T_S_nom`, 12  
 nomsolirradiance `_S_S_nom`, 12  
 nomsolluminosity `_L_S_nom`, 12  
 nomsolmassparam `_GM_S_nom`, 12  
 nomsolradius `_R_S_nom`, 12  
 nomterreqradius `_Re_E_nom`, 12  
 nomterrmasparam `_GM_E_nom`, 12  
 nomterrpolarradius `_Rp_E_nom`, 12  
 nuclearmagneton , 10  
  
 ohm `_Ohm`, 11  
 ounce `_oz`, 16  
  
 parsec `_pc`, 13  
 partsperbillion `_ppb`, 14  
 partspermillion `_ppm`, 14  
 partsperquadrillion `_ppq`, 14  
 partspertrillion `_ppt`, 14  
 pascal `_Pa`, 11  
 percent `_percent`, 14  
 permille `_permille`, 14  
 pi , 10  
 pica `_pica`, 15  
 pint `_pint`, 15  
 planckconstant , 10  
 point `_pt`, 15  
 poiseuille `_Pl`, 14  
 pound `_lb`, 16  
 protonmagneticmoment , 10  
 protonmass , 10  
  
 quart `_qt`, 15  
 quarter `_qtr`, 16  
  
 rad `_Rad`, 14  
 radian `_rad`, 11  
 reducedplanckconstant , 10  
  
 rem `_rem`, 14  
 rod `_rd`, 15  
 rydbergconstant , 10  
  
 siemens `_S`, 11  
 sievert `_Sv`, 11  
 spat `_sp`, 13  
 speedoflight , 10  
 standardatmosphere `_atm`, 13  
 standardgravity , 10  
 stefanboltzmannconstant , 10  
 steradian `_sr`, 11  
 stone `_st`, 16  
 svedberg `_svedberg`, 13  
  
 technicalatmosphere `_at`, 13  
 tesla `_T`, 11  
 thermochemicalcalorie `_cal`, 14  
 thou `_th`, 15  
 tonne `_t`, 12  
 tonoftnt `_t_TNT`, 14  
 torr `_Torr`, 13  
 turn `_tr`, 13  
  
 uscup `_cup_US`, 17  
 usfluid dram `_fl_dr_US`, 17  
 usfluidounce `_fl_oz_US`, 17  
 usgallon `_gal_US`, 17  
 usgill `_gi_US`, 17  
 ushundredweight `_cwt_US`, 17  
 uspint `_pint_US`, 17  
 usquart `_qt_US`, 17  
 usquarter `_qtr_US`, 17  
 ussurveyacre `_ac_US`, 17  
 ussurveycable `_cbl_US`, 17  
 ussurveychain `_ch_US`, 17  
 ussurveyfathom `_ftm_US`, 17  
 ussurveyfoot `_ft_US`, 17  
 ussurveyfurlong `_fur_US`, 17  
 ussurveyhand `_hh_US`, 17  
 ussurveyinch `_in_US`, 17  
 ussurveyleague `_lea_US`, 17  
 ussurveylink `_li_US`, 17  
 ussurvey mile `_mi_US`, 17  
 ussurveyrod `_rd_US`, 17  
 ussurveyyard `_yd_US`, 17

ustablespoon \_Tbsp\_US, 17  
usteaspoon \_tsp\_US, 17  
uston \_ton\_US, 17  
  
vacuumpermeability , 10  
vacuumpermittivity , 10  
volt \_V, 11  
voltampere \_VA, 14

watt \_W, 11  
watthour \_Wh, 14  
wattsecond \_Ws, 14  
weber \_Wb, 11  
week \_wk, 13  
  
yard \_yd, 15  
year \_a, 13

## Index of Currencies

AfghanAfghani \_AFN, 18  
AfghanPul \_cAFN, 18  
AlbanianLek \_ALL, 18  
AlgerianDinar \_DZD, 19  
AlgerianSanteem \_cDZD, 19  
AngolanCentimo \_cAOA, 18  
AngolanKwanza \_AOA, 18  
ArgentineCentavo \_cARS, 18  
ArgentinePeso \_ARS, 18  
ArmenianDram \_AMD, 18  
ArmenianLuma \_cAMD, 18  
ArubanCent \_cAWG, 26  
ArubanFlorin \_AWG, 26  
AustralianCent \_cAUD, 18  
AustralianDollar \_AUD, 18  
AzerbaijaniManat \_AZN, 18  
AzerbaijaniQepik \_cAZN, 18

BahamianCent \_cBSD, 26  
BahamianDollar \_BSD, 26  
BahrainiDinar \_BHD, 26  
BahrainiFils \_cBHD, 26  
BangladeshiPoisha \_cBDT, 18  
BangladeshiTaka \_BDT, 18  
BarbadianCent \_cBBD, 26  
BarbadianDollar \_BBD, 26  
BelarusianKapiejka \_cBYN, 18  
BelarusianRuble \_BYN, 18  
BelizeCent \_cBZD, 26  
BelizeDollar \_BZD, 26  
BermudianCent \_cBMD, 26  
BermudianDollar \_BMD, 26  
BhutaneseChhertum \_cBTN, 26  
BhutaneseNgultrum \_BTN, 26  
BolivianBoliviano \_BOB, 18  
BolivianCentavo \_cBOB, 18  
BosnianFenings \_cBAM, 18  
BosnianMark \_BAM, 18  
BotswanaPula \_BWP, 18  
BotswanaThebe \_cBWP, 18  
BrazilianCentavo \_cBRL, 18  
BrazilianReal \_BRL, 18

BruneiDollar \_BND, 26  
BruneiSen \_cBND, 26  
BulgarianLev \_BGN, 26  
BulgarianStotinka \_cBGN, 26  
BurmeseKyat \_MMK, 22  
BurmesePya \_cMMK, 22  
BurundianCentime \_cBIF, 18  
BurundianFranc \_BIF, 18

CambodianRiel \_KHR, 21  
CanadianCent \_cCAD, 19  
CanadianDollar \_CAD, 19  
CapeVerdeanCentavo \_cCVE, 27  
CapeVerdeanEscudo \_CVE, 27  
CaymanIslandsCent \_cKYD, 27  
CaymanIslandsDollar \_KYD, 27  
CentralAfricanCFACentime \_cXAF,  
28  
CentralAfricanCFAFranc \_XAF, 28  
CFPCentime \_cXPF, 29  
CFPFranc \_XPF, 29  
ChileanCentavo \_cCLP, 19  
ChileanPeso \_CLP, 19  
ChineseRenminbiFen \_cCNY, 19  
ChineseRenminbiYuan \_CNY, 19  
ColombianCentavo \_cCOP, 19  
ColombianPeso \_COP, 19  
ComorianCentime \_cKMF, 27  
Comorianfranc \_KMF, 27  
CongolesCentime \_cCDF, 19  
CongolesFranc \_CDF, 19  
CostaRicanCentimos \_cCRC, 19  
CostaRicanColon \_CRC, 19  
CroatianKuna \_HRK, 20  
CroatianLipa \_cHRK, 20  
CubanCentavo \_cCUP, 26  
CubanoConvertibleCentavo \_cCUC,  
26  
CubanoConvertiblePeso \_CUC, 26  
CubanPeso \_CUP, 26  
CzechHaler \_cCZK, 19  
CzechKoruna \_CZK, 19

DanishKrone \_DKK, 19  
 DanishOre \_cDKK, 19  
 DjiboutianCentime \_cDJF, 27  
 DjiboutianFranc \_DJF, 27  
 DominicanCentavo \_cDOP, 19  
 DominicanPeso \_DOP, 19  
  
 EasternCaribbeanCent \_cXCD, 28  
 EasternCaribbeanDollar \_XCD, 28  
 EgyptianPiastre \_cEGP, 19  
 EgyptianPound \_EGP, 19  
 EritreanCent \_cERN, 27  
 EritreanNakfa \_ERN, 27  
 EthiopianBirr \_ETB, 19  
 EthiopianSantim \_cETB, 19  
  
 FalklandIslandsPenny \_cFKP, 27  
 FalklandIslandsPound \_FKP, 27  
 FijianCent \_cFJD, 19  
 FijianDollar \_FJD, 19  
  
 GambianButut \_cGMD, 20  
 GambianDalasi \_GMD, 20  
 GeorgianLari \_GEL, 20  
 GeorgianTetri \_cGEL, 20  
 GhanaianCedi \_GHS, 20  
 GhanaianPesewa \_cGHS, 20  
 GibraltarPenny \_cGIP, 27  
 GibraltarPound \_GIP, 27  
 GuatemalanCentavo \_cGTQ, 20  
 GuatemalanQuetzal \_GTQ, 20  
 GuernseyPenny \_cGGP, 27  
 GuernseyPound \_GGP, 27  
 GuineanCentime \_cGNF, 20  
 GuineanFranc \_GNF, 20  
 GuyaneseCent \_cGYD, 20  
 GuyaneseDollar \_GYD, 20  
  
 HaitianCentime \_cHTG, 20  
 HaitianGourde \_HTG, 20  
 HonduranCentavo \_cHNL, 20  
 HonduranLempira \_HNL, 20  
 HongKongCent \_cHKD, 20  
 HongKongDollar \_HKD, 20  
 HungarianFiller \_cHUF, 20  
 HungarianForint \_HUF, 20  
  
 IcelandicKrona \_ISK, 21  
 IndianPaisa \_cINR, 21  
 IndianRupee \_INR, 21  
 IndonesianRupiah \_IDR, 20  
 IndonesianSen \_cIDR, 20  
 IranianRial \_IRR, 21  
 IranianToman \_cIRR, 21  
 IraqiDinar \_IQD, 21  
 IraqiFils \_cIQD, 21  
 IsraeliNewAgora \_cILS, 20  
 IsraeliNewShekel \_ILS, 20  
  
 JamaicanCent \_cJMD, 21  
 JamaicanDollar \_JMD, 21  
 JapaneseYen \_JPY, 21  
 JerseyPenny \_cJEP, 27  
 JerseyPound \_JEP, 27  
 JordanianDinar \_JOD, 27  
 JordanianFils \_cJOD, 27  
  
 KazakhstaniTenge \_KZT, 21  
 KazakhstaniTiyn \_cKZT, 21  
 KenyanCent \_cKES, 21  
 KenyanShilling \_KES, 21  
 KiribatiCent \_cKID, 27  
 KiribatiDollar \_KID, 27  
 KuwaitiDinar \_KWD, 21  
 KuwaitiFils \_cKWD, 21  
 KyrgyzstaniSom \_KGS, 21  
 KyrgyzstaniTiyn \_cKGS, 21  
  
 LaoAtt \_cLAK, 21  
 LaoKip \_LAK, 21  
 LebanesePound \_LBP, 27  
 LebaneseQeresh \_cLBP, 27  
 LiberianCent \_cLRD, 21  
 LiberianDollar \_LRD, 21  
 LibyanDinar \_LYD, 22  
 LibyanDirham \_cLYD, 22  
  
 MacaneseAvo \_cMOP, 27  
 MacanesePataca \_MOP, 27  
 MacedonianDenar \_MKD, 22  
 MacedonianDeni \_cMKD, 22  
 MalagasyAriary \_MGA, 22  
 MalagasyIraimbilanja \_cMGA, 22



MalawianKwacha \_MWK, 22  
 MalawianTambala \_cMWK, 22  
 MalaysianRinggit \_MYR, 22  
 MalaysianSen \_cMYR, 22  
 MaldivianLaari \_cMVR, 22  
 MaldivianRufiyaa \_MVR, 22  
 ManxPenny \_cIMP, 27  
 ManxPound \_IMP, 27  
 MauritanianKhoums \_cMRU, 22  
 MauritanianOuguiya \_MRU, 22  
 MauritianCent \_cMUR, 22  
 MauritianRupee \_MUR, 22  
 MexicanCentavo \_cMXN, 22  
 MexicanPeso \_MXN, 22  
 MoldovanBan \_cMDL, 22  
 MoldovanLeu \_MDL, 22  
 MongolianMonggo \_cMNT, 22  
 MongolianTogrog \_MNT, 22  
 MoroccanDirham \_MAD, 22  
 MoroccanSantim \_cMAD, 22  
 MozambicanCentavo \_cMZN, 22  
 MozambicanMetical \_MZN, 22  
  
 NamibianCent \_cNAD, 28  
 NamibianDollar \_NAD, 28  
 NepalesePaissa \_cNPR, 28  
 NepaleseRupee \_NPR, 28  
 NetherlandsAntilleanCent \_cANG, 26  
 NetherlandsAntilleanGuilder \_ANG,  
 26  
 NewTaiwanCent \_cTWD, 25  
 NewTaiwanDollar \_TWD, 25  
 NewZealandCent \_cNZD, 23  
 NewZealandDollar \_NZD, 23  
 NicaraguanCentavo \_cNIO, 23  
 NicaraguanCordoba \_NIO, 23  
 NigerianKobo \_cNGN, 23  
 NigerianNaira \_NGN, 23  
 NorthKoreanChon \_cKPW, 21  
 NorthKoreanWon \_KPW, 21  
 NorwegianKrone \_NOK, 23  
 NorwegianOre \_cNOK, 23  
  
 OmaniBaisa \_cOMR, 28  
 OmaniRial \_OMR, 28  
 PakistaniPaissa \_cPKR, 23  
 PakistaniRupee \_PKR, 23  
 PanamanianBalboa \_PAB, 28  
 PanamanianCentesimo \_cPAB, 28  
 PapuaNewGuineanKina \_PGK, 23  
 PapuaNewGuineanToea \_cPGK, 23  
 ParaguayanCentimo \_cPYG, 23  
 ParaguayanGuarani \_PYG, 23  
 PennySterling \_cGBP, 20  
 PeruvianCentimo \_cPEN, 23  
 PeruvianSol \_PEN, 23  
 PhilippinePeso \_PHP, 23  
 PhilippineSentimo \_cPHP, 23  
 PolishGrosz \_cPLN, 23  
 PolishZloty \_PLN, 23  
 PoundSterling \_GBP, 20  
  
 QatariDirham \_cQAR, 23  
 QatariRiyal \_QAR, 23  
  
 RomanianBan \_cRON, 23  
 RomanianLeu \_RON, 23  
 RussianKopeyka \_cRUB, 23  
 RussianRuble \_RUB, 23  
 RwandanCentime \_cRWF, 24  
 RwandanFranc \_RWF, 24  
  
 SaintHelenaPenny \_cSHP, 28  
 SaintHelenaPound \_SHP, 28  
 SamoanSene \_cWST, 25  
 SamoanTala \_WST, 25  
 SaoTomeAndPrincipeCentimo  
 \_cSTN, 28  
 SaoTomeAndPrincipeDobra \_STN,  
 28  
 SaudiHalalah \_cSAR, 28  
 SaudiRiyal \_SAR, 28  
 SerbianDinar \_RSD, 23  
 SerbianPara \_cRSD, 23  
 SeychelloisCent \_cSCR, 24  
 SeychelloisRupee \_SCR, 24  
 SierraLeoneanCent \_cSLL, 24  
 SierraLeoneanLeone \_SLL, 24  
 SingaporeCent \_cSGD, 24  
 SingaporeDollar \_SGD, 24  
 SolomonIslandsCent \_cSBD, 24  
 SolomonIslandsDollar \_SBD, 24

SomalilandCent \_cSQS, 24  
 SomalilandShilling \_SQS, 24  
 SomaliSenti \_cSOS, 24  
 SomaliShilling \_SOS, 24  
 SouthAfricanCent \_cZAR, 25  
 SouthAfricanRand \_ZAR, 25  
 SouthKoreanJeon \_cKRW, 21  
 SouthKoreanWon \_KRW, 21  
 SouthSudanesePiaster \_cSSP, 28  
 SouthSudanesePound \_SSP, 28  
 SriLankanCent \_cLKR, 21  
 SriLankanRupee \_LKR, 21  
 SudanesePound \_SDG, 24  
 SudaneseQirsh \_cSDG, 24  
 SurinameseCent \_cSRD, 24  
 SurinameseDollar \_SRD, 24  
 SwaziCent \_cSZL, 28  
 SwaziLilangeni \_SZL, 28  
 SwedishKrona \_SEK, 24  
 SwedishOre \_cSEK, 24  
 SwissFranc \_CHF, 19  
 SwissRappen \_cCHF, 19  
 SyrianPiastre \_cSYP, 24  
 SyrianPound \_SYP, 24  
  
 TajikistaniDram \_cTJS, 24  
 TajikistaniSamani \_TJS, 24  
 TanzanianSenti \_cTZS, 25  
 TanzanianShilling \_TZS, 25  
 ThaiBaht \_THB, 24  
 ThaiSatang \_cTHB, 24  
 Tonganpaanga \_TOP, 24  
 TonganSeniti \_cTOP, 24  
 TransnistrianKopeck \_cPRB, 28  
 TransnistrianRuble \_PRB, 28  
 TrinidadAndTobagoCent \_cTTD, 25  
 TrinidadAndTobagoDollar \_TTD, 25  
  
 TurkishKurus \_cTRY, 25  
 TurkishLira \_TRY, 25  
 TurkmenistanManat \_TMT, 28  
 TurkmenistanTenge \_cTMT, 28  
 TuvaluanCent \_cTVD, 28  
 TuvaluanDollar \_TVD, 28  
  
 UgandanCent \_cUGX, 25  
 UgandanShilling \_UGX, 25  
 UkrainianHryvnia \_UAH, 25  
 UkrainianKopiyka \_cUAH, 25  
 UnitedArabEmiratesDirham \_AED,  
 26  
 UnitedArabEmiratesFils \_cAED, 26  
 UruguayanCentesimo \_cUYU, 25  
 UruguayanPeso \_UYU, 25  
 USCent \_cUSD, 25  
 USDollar \_USD, 25  
 UzbekistaniSom \_UZS, 25  
 UzbekistaniTiyin \_cUZS, 25  
  
 VenezuelanBolivarSoberano \_VES,  
 25  
 VenezuelanCentimoSoberano \_cVES,  
 25  
 VietnameseDong \_VND, 25  
 VietnameseXu \_cVND, 25  
  
 WestAfricanCFACentime \_cXOF, 29  
 WestAfricanCFAFranc \_XOF, 29  
  
 YemeniDinar \_cYER, 25  
 YemeniRial \_YER, 25  
  
 ZambianKwacha \_ZMW, 25  
 ZambianNgwee \_cZMW, 25  
 ZimbabweanBonds \_ZWL, 29  
 ZimbabweanCent \_cZWL, 29