

RedCrab

Calculator V

User Manual

Copyright © by RedCrab, 2009 - 2015

RedCrab Calculator

Version 5.3

This program is shareware. Without valid shareware license, you can use the program in demo mode with limited function. Full functionality is available with the purchase of a shareware license available.

In shareware mode you can load data sheet which includes shareware programs features. The worksheet is free to use, but the program code is read only and file saved is disabled.

Copyright

Software and manual are copyright. You can copy and pass it, but it is forbidden to change the software or the manual.

We are not liable for any error in software or manual. Usage is at your own risk.

System requirement

Operating system: *Microsoft Windows* Vista, W7, W8.x, W10

Framework 4.5 or higher

RedCrab is portable. No installation of the software is required. You can just copy the software to your system and starts the programs. RedCrab can run on removable drives like DVD or USB-Stick.

Calculation range and accuracy:

Floating point

Accuracy: 15 – 16 digits
Calculation range: $\pm 5 \times 10^{-324}$ to $\pm 1.7 \times 10^{308}$

Hexadecimal: 12 digits

Floating point is due to the large range of values suitable for engineering and scientific calculations.

Decimal

Accuracy: 28-29 digits

Calculation range: from -79,228,162,514,264,337,593,543,950,335
to 79,228,162,514,264,337,593,543,950,335

Is approximate -7.9×10^{28} to 7.9×10^{28}

Hexadecimal: 16 digits

Decimal is suitable due to the small rounding error for financial and monetary calculations.

***Windows** is a registered trademark of Microsoft Corporation. All other trademarks are the property of their respective owners.

Contents

- 1.0 Mathematical Expressions
 - 1.1 Basics
 - 1.2 Start Calculation
 - 1.3 Simple Addition
 - 1.4 Exponent
 - 1.5 Subscript
 - 1.6 Alternative Font
 - 1.7 Implied Multiplication
 - 1.8 Fractions
 - 1.9 Root
 - 1.10 Hexadecimal, Octal and Binary Input
 - 1.11 Operators
 - 1.12 Variable Overload
 - 1.13 Data Fields
 - 1.14 Multidimensional Fields
 - 1.15 Work with Fields
 - 1.16 Define a mathematical Function
 - 1.17 Scope of Function Parameters
 - 1.18 Calculate Selected Formulas
 - 1.19 Units of Measurement
 - 1.20 List of Units of Measurement
- 2.0 Result Formatting
 - 2.1 Result Mode Prefix
 - 2.2 Specification of a Prefix
 - 2.3 Number of Decimal Places
 - 2.4 Formatting of Results
 - 2.5 Display of Units of Measurement
 - 2.6 Result speech (Narrator)
 - 2.7 Formatting Tables
 - 2.8 Edit Formatting
 - 2.9 Narrator settings
 - 2.10 Display Tables
- 3.0 Display Results Graphically with Charts
 - 3.1 Chart Type
 - 3.2 Legend Settings
 - 3.3 Axis Settings
 - 3.4 Chart Options
 - 3.5 Print Chart box

- 4.1 Insert Text Box
- 4.2 Insert Pictures
- 4.3 Program Box
- 4.4 Insert Slider
- 4.5 Slider Range Selection
- 4.6 Insert Label

- 5.0 Toolbar **Tools**
- 5.1 Page Lock
- 5.2 Cell Unlock
- 5.3 Remark
- 5.4 Autocalc
- 5.5 Tooltip Language
- 5.6 Keyboard Settings
- 5.7 Error Messages

- 6.0 Functions and operators

- 6.1 Standard Functions
Abs, Ceil, DTime, DTimef, Floor, Frac, Int, Rnd, Round, Sign, Sqr, Sqrt, URnd

- 6.2 Scientific Functions
ACos, ASin, ATan, Cos, Cosh, Cot, Deg, Exp, Ln, Log, Log2, Log8, Log16, Rad, Sin, Sinh, Tan Tanh, Ld, Lg, Log10

- 6.3 Programmer Functions and Operators
And, Div, Excl, Incl, Mod, Not, Or, Shl, Shr, Xor

- 6.4 Data Field Functions
Aver, Cols, Count, Diff, Dim, Fill, Join, Maxi, Mini, Patt, Rows

- 6.5 Matrix Functions
Det, Invx, Mulx, Trans

- 6.6 Statistics Functions
Cusum, DSort, LQuart, Mean, Median, Prod, Qran, Sort, SStDev, StDev, Sum, SVari, UQuart, Vari

- 6.7 Financial Functions
FDDB, FFV, FIPmt, FIRR, FMIRR, FNPer, FNPV, FPmt, FPPmt, FPV, FRate, FSLN, FSYD

7.0 View Menu (Toolbar)

7.1 License Activation

Attachment

Keyboards

RedCrab – The Calculator

RedCrab is a math program with a full screen editor. The editor allows the free placement of algebraic formulas on the worksheet in mathematical notation. Mathematical symbols like fraction lines and roots are supported.

The handling of the basic functions is just like a conventional calculator. There is no training required.

This guide describes advanced features which a normal calculator does not possess. Additional **RedCrab's** menu elements have tool tips with examples in English and German language.

Additional Information: www.redchillicrab.com/en/redcrab/tutor.html

RedCrab is fully portable. The program can be started from external data storage source without installation. Settings will be stored as a file in the RedCrab start directory **Tools\Settings\user.config**.

1.0 Mathematical Expressions

1.1 Basics

You can write your formula basically at any editor position. Any expression may occupy any number of rows and columns. It is not allowed splitting an expression and to continue in the next row.

Wrong: $z = 12+14+15+20$
 $+5+10$

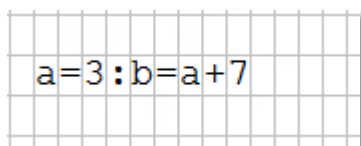
Correct: $z = 12+14+15+20+5+10$

Correct: $X = 12+14+15+20$
 $Z = X+5+10$

You can write several mathematical expressions on one work sheet. The expressions result displays only if terminated with equal sign.

Example 1: $a+b$ $=108$
 $a=27+9$
 $8*4$ $=32$
 $b=12*6$ $=72$

Several mathematical expressions can be written per row. Between each mathematical expression, there must either a minimum number at four blank columns or a colon must be set.



Example 1:

Example 2:

a=3	b=a+7
-----	-------

$C1 = \frac{1}{2\pi f_H Z\sqrt{2}}$	$= 5.024 \cdot 10^{-6}$
$L1 = \frac{Z\sqrt{2}}{2\pi f_H}$	$= 643.1 \cdot 10^{-6}$
$C2 = C1$	$= 5.024 \cdot 10^{-6}$
$L2 = L1$	$= 643.1 \cdot 10^{-6}$
$C3 = \frac{1}{2\pi f_L Z\sqrt{2}}$	$= 17.58 \cdot 10^{-6}$

An equal sign behind a formula is always assigned to the previous formula, even if the distance to the formula is greater than the column space setting. In the example right, the distance of the equal sign is up to eight columns, although the minimum distance is only four columns.

Close proximity can caused unexpected errors. For error localization **RedCrab** marked the cell where an error is detected with a blue frame. It also marks the incorrect formula with a red frame. In the example below, an invalid assignment is signaled. The red box shows, however, that two formulas were joined because the distance is too close. The setting in this example is 4 columns; the distance between the formulas is only 2 columns.

$Q_s = \frac{2\pi f_0 L}{R} = L = 2.5 \cdot 10^{-3}$
--

Error : Expression not applicable on this position

1.2 Start Calculation

Start the calculation with a click on the Enter button or press **F8** or **Ctrl + Enter** keys.

Reset clears the displayed results. Clear clears the worksheet complete. The Stop symbol terminated a running calculation. But this is only in the calculation of large data fields of importance.



1.3 Simple Addition

1. Enter the expression $17 + 4$
2. For result press **Ctrl + Enter**

The **Ctrl + Enter** key starts **RedCrab** and displays the result. Alternative click the function panels **Enter** button. Results are always displayed in blue.

The display shows: $17 + 4 = 21$

Variable and Values

1. Enter the expression $17 + 4 + X$
2. Enter the assignment $X = 43$
3. For result press **Ctrl + Enter**

RedCrab displays the result: 64

The display shows: $17 + 4 + X = 64$
 $X = 43$

The assignment can be entered at any position.

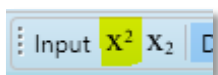
1.4 Exponent

The expression: $c = a^2 + 4^2$.

1. Enter the expression: $c = a$ **Ctrl + 2** $+ 4$ **Ctrl + 2** $=$
2. Press **Ctrl + Enter** to display result.

The display shows: $c = 3^2 + 4^2 = 25$

The keys **Ctrl+2** write the exponent 2. With the keys **Ctrl+3** you can write the exponent 3.



For use of any other values for exponents, press the **Ctrl+6** keys or click the **Superscript** Button to enter the **Super** mode. Then enter the exponent value. Press **Ctrl+6** or **Enter** or click the **Superscript** Button to leave the super mode.

Key functions:

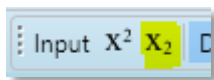
- Function key **Ctrl + 6** enabled / disabled **Superscript** mode
- Function key **F3** enabled / disabled **Superscript** mode.
- The **Enter** or the '=' key leaves the Superscript mode.

If you activate Superscript when the cursor is over a character, or a range is selected, the character under the cursor or the selected range changed from normal letters in superscript. The **Superscript** mode is not enabled in this case, only the sign is changed. Similarly, the character can be reset by **Superscript** in normal font.

1.5 Subscript

Enter the formula: $X_L = 2 * 628$

Press the keys: X **Ctrl**+ _ L **Enter** = 2 * 6 2 8 =



With the keys **Ctrl**+ _ (*underscore*) you can switch **Subscript** on / off. Alternative you can use **Enter** to leave the **Subscript** region.

Key functions:

- The *underscore*_ key and **Ctrl** + _ (*underscore*) toggles **Subscript** too.
- **Subscript** mode can be enabled / disabled using the function key **F4**.
- The **Enter** or the '=' key leaves the **Subscript** mode.

If you activate **Subscript** when the cursor is over a character, or a range is selected, the character under the cursor or the selected range changed from normal letters in subscript. The **Subscript** mode is not enabled in this case, only the sign is changed. Similarly, the character can be reset by **Subscript** in normal font.

Instead of using the keyboard, you can activate the subscript mode with a click on the **Subscript** button on the **Input** toolbar.

1.6 Alternative Font

Enter the formula: $\omega = 2 * \pi * f$

The **Ctrl** key shifts the letters to the alternative font. The example above shows that the keys **Ctrl + P** displayed the Hellenic letter **Pi** (π).

For a complete list of special symbols, you can refer to the description of the keyboard below. If you work with RedCrab, the simplest way is to open the virtual keyboard with a click on the **Virtual keyboard** button on the **Tools** menu ribbon.

1.7 Implied Multiplication

Enter the formula: $\omega = 2 \pi f$

The example above show one more features of RedCrab: the **implied multiplication**. That means you do not need to include the multiplication operator. Example: **RedCrab** interprets $\omega = 2 \pi f$ as $\omega = 2 * \pi * f$

A space is required between the names of the variables. Related letters are interpreted as one word.

Example:	a b c	is equivalent to	$a * b * c$
	3 a b	is equivalent to	$3 * a * b$
	2X _L	is equivalent to	$2 * X_L$
	R ₁ R ₂	is equivalent to	$R_1 * R_2$

1.8 Fractions

Entering a fraction line: Press the **Slash** key (/) two times and a three-character fraction bar will be displayed. By repeatedly pressing the key the fraction bar is extended by one character. In general, it is sufficient if you continue entering data above and below the fracture line. When typing the numerator or denominator data, the fraction bar is automatically extended by the editor as far as it is required.

If you have taken the fraction line, the cursor is in the first column after the line. Press in this position **Enter** key, the cursor moves over the slash to the first position of the numerator. After entering the numerator, press again **Enter**, the

cursor jump to the first position of the Denominators. After entering the data, press **Enter** again. The cursor jumps back into the column right of the fraction line.

! The fraction bar must exceed at least 1 character front and rear.

Examples:

$\frac{123}{abc}$ wrong

$\frac{123}{Abc}$ correct

The display shows:
$$f = \frac{1}{2\pi\sqrt{LC}} = 2.6 \cdot 10^3$$

$$L = 0.8 \cdot 10^{-3}$$

$$C = 4.7 \cdot 10^{-6}$$

1.9 Root

Set the root character with the keys **CTRL+I** to the desired position. Then mark the area which is to be included under the root. Finally set the cursor on the root of character, the editor draws the root symbol over the marked area.

For one-line root calculation, the following steps apply:

1. Set root symbol with **CTRL+I**.
2. Enter the data
3. Holding down the Shift key and with **Cursor-left** key reposition to the root sign.

The editor draws the root symbol over the marked area.

For multi-line data in the root (e.g., fractions):

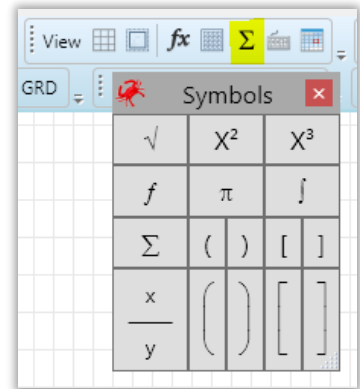
1. Set root symbol with **CTRL+I**.
2. Data entry.
3. Mark the area for the root with the mouse.
4. Click the mouse on the root symbol.

The editor draws the root symbol over the marked area.

In order to highlight the area, it is sufficient if the last column under the root is marked.

To change the area under the root, highlight, as described above, the new field and then click the cell of the root sign. The roots then marked the new area.

By double-clicking on the root symbol the root lines around the data is removed.



Instead per keyboard you can insert the special symbols per mouse click. To do that, open the **Symbol** panel with a click on the **Tools** menu ribbons **Symbol pad** button.

1.10 Hexadecimal, Octal and Binary Input

The **RedCrab** editor accepts input of hexadecimal numbers up to 13 digits. The hexadecimal number must mark with a dollar symbol before it. The use of small or capital letters are allowed.

Example: \$1F2A or \$1f2a

An octal number is marked with the dollar symbol and the letters **oct**.

Example: \$oct3721

A binary number is marked with the dollar symbol and the letters **bin**.

Example: \$bin110101

You can use hexadecimal, octal or binary numbers in any position of a formula like decimal numbers. Between this number and the following number or variable must be a space or operator symbol.

Example: Correct: \$1F2A*X or \$1F2A X

Wrong: \$1F2AX => generate an error message.

1.11 Operators

RedCrab enable you to enter numbers and functions in a simple, straightforward sequence. The table below shows the order in which functions in expressions are entered and evaluated.

- 1 SIN(), NOT(), root... and all functions left of the argument
- 2 X^2 , .. ,
- 3 join
- 4 *, /, DIV, MOD, AND, SHL, SHR, INCL, EXCL,
- 5 +, -, OR, XOR

Within a priority group, **RedCrab** evaluates functions from left to right. Calculations within a pair of parentheses are evaluated first.

1.12 Variable Overload

You can assign different values to the same variable.

Example: $P = U \cdot I =$
 $P = U^2 / R =$

But an overloaded variable has no defined value and can't be used for further calculations or result boxes.

Overloaded constants can be reused. Example: the constant e is occupied by the **Euler** number $e = 2.7182818$. You can overload this value and use e for further calculations.

Example 1: $x=e=2.7182818$

Example 2: $e=11$
 $X=2e=22$

Predefined Values

e	Eulerscher Number: 2.7182818284590452...
π	PI: 3.1415....
TRUE	1
FALSE	0
NIL	undefined
IPRE	360

1.13 Data Fields

The following section describes how to work with dynamic data fields. **RedCrab** can manage multi-dimensional fields. Size and dimensions are limited by the resources of the computer only.

The handling of the fields corresponds to the simple variables. That means no special declaration of variables is required. To generate a field, a sequence of numbers is assigned to a variable. The sequence is written in square brackets and separated by commas.

Example: $x = [1, 3, 7, 12]$

The assignment of a series shows the following example. It will be assigned to the variable x 180 indices with the values 1 to 180.

Example: $x = [1..180]$

A series is always expanded in increments of + /- 1. Other step sizes can multiply or divide by the field generated, or in definition of data fields you can optionally specify the increment of a range (example 2).

Example 1: $x = 5[0..4] = 0 \quad 5 \quad 10 \quad 15 \quad 20$
 $x = [0..5]/5 = 0 \quad 0.2 \quad 0.4 \quad 0.6 \quad 0.8 \quad 1$
 $x = 5/[1..5] = 5 \quad 2.5 \quad 1.67 \quad 1.25 \quad 1$
 $x = 2[5..0] = 10 \quad 8 \quad 6 \quad 4 \quad 2 \quad 0$

Example 2: $x = [2..5:0.75] = 2 \quad 2.75 \quad 3.5 \quad 4.25 \quad 5$

Series, individual values and variables can be combined.

Example: $x = [1, 5 \dots 8, 12, 15] = 1 \quad 5 \quad 6 \quad 7 \quad 8 \quad 12 \quad 15$

Example: $a = 3$
 $b = 12$
 $x = [1, a \dots 5, b] = 1 \quad 3 \quad 4 \quad 5 \quad 12$

Fields are treated as normal values in calculations and can be combined with all operators and functions. The result is a field as well.

Example: $[2, 4, 7] + 10 = 12 \quad 14 \quad 17 \quad (2+10 \quad 4+10 \quad 7+10)$

Example: $\sin([30, 60, 90]) = 0.5 \quad 0.87 \quad 1$

Example: $[12, 18, 36, 44] \bmod 10 = 2 \quad 8 \quad 6 \quad 4$

Example: $C = 4.6 \cdot 10^{-6}$
 $f = [1200, 1600, 2000, 2600]$

$$16,9 \quad 13 \quad \frac{1}{2\pi f C} \quad X_c = \quad = 28.2 \quad 21.2$$

The example above shows a list as a result, which contains four different values of f .

Individual components of a field can be accessed via the index.

Example: $x = [11 \dots 20]$
 $y = x[1, 4, 6 \dots 8] = 11 \quad 14 \quad 16 \quad 17 \quad 18$

1.14 Multidimensional Fields

To generate multi-line fields, separate each row by semicolon.

Example: $x = [1, 2, 3; 4, 5, 6] = \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{matrix}$

If rows have different length, the missing indexes are filled with zeros.

Example: $x = [1 \dots 5; 2, 4, 6; 3 \dots 9] = \begin{matrix} 1 & 2 & 3 & 4 & 5 & 0 & 0 \\ 2 & 4 & 6 & 0 & 0 & 0 & 0 \\ 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{matrix}$

Fields with three rows can be written alternative with a large bracket.

Example: $x = \begin{bmatrix} 1, 2, 3 \\ 4, 5, 6 \\ 7, 8, 9 \end{bmatrix} = \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$

This standard is generally used in matrix notation, but has no effect on the following calculations. For multiplication of matrices, refer to section below ***Mulx*** function.

By entering the data, as described above, one-and two-dimensional fields are generated. Fields with three or more dimensions can be generated computationally.

1.15 Work with Fields

Two fields can be operands of a mathematical expression when the fields are of the same type. This means they must have the same size and number of dimensions. An exception is different length in the first dimension. The excess of the longer field are ignored.

Example: $a = [2, 3, 4, 5]$
 $b = [10, 11, 12, 13]$
 $c = a + b = 12 \ 14 \ 16 \ 18 \quad (2+10 \ 3+11 \ 4+12 \ 5+13)$

Example: $a = [2, 3, 4, 5]$
 $b = [10, 11, 12, 13, 14, 15]$
 $c = a + b = 12 \ 14 \ 16 \ 18$

Excess field length of b (14,15) is ignored.

Example: $a = [2..5; \ 20..23]$
 $b = [10..13; \ 30..33]$
 $c = a + b = 12 \ 14 \ 16 \ 18$
 $50 \ 52 \ 54 \ 56$

Example: $a = [2..5; \ 20..23]$
 $b = [10..13; \ 30..33; \ 40, 44, 45, 48]$
 $c = a + b = 12 \ 14 \ 16 \ 18$
 $50 \ 52 \ 54 \ 56$

In this example, the third row of b is ignored

Example: $a = [2..5; \ 20..23]$
 $b = [10..13; \ 30..33; \ 40, 44, 45, 48]$
 $c = a + b[1, 3] = 12 \ 14 \ 16 \ 18$
 $60 \ 65 \ 67 \ 71$

In this example, a from row 1 is added with b from row 3

In the examples above, each index of a is added with the corresponding index of b . Alternatively **RedCrab** can calculate fields in which each index of a field a is calculated with each index of the field b . The result is a multidimensional field of the size indices a times indices b .

The empty brackets following c declares the result as a multidimensional field and determines the type of the following calculation.

Example: $a = [10, 15]$
 $b = [2..4]$
 $c[] = a+b = \begin{matrix} 12 & 13 & 14 & (10+2 & 10+3 & 10+4) \\ 17 & 18 & 19 & (15+2 & 15+3 & 15+4) \end{matrix}$

Example: $a = [3..6]$
 $b = [11..15]$
 $c[] = a*b = \begin{matrix} & 33 & 36 & 39 & 42 & 45 \\ 44 & 48 & 52 & 56 & 60 \\ 55 & 60 & 65 & 70 & 75 \\ 66 & 72 & 78 & 84 & 90 \end{matrix}$

The next example shows to multiply a one-dimensional field by a two-dimensional field. The result is a three-dimensional field.

Example: $a = [3..6]$
 $b = [11..15]$
 $c[] = a * b$
 $d[] = a * c = \begin{matrix} & 99 & 108 & 117 & 126 & 135 \\ 132 & 144 & 156 & 168 & 180 \\ 165 & 180 & 195 & 210 & 225 \\ 198 & 216 & 234 & 252 & 270 \end{matrix}$

The display above shows the two-dimensional field of the first level. This is the field that lies behind the first row. Other fields can be accessed via index.

Example: $d[2] = \begin{matrix} 132 & 144 & 156 & 168 & 180 \\ 176 & 195 & 208 & 224 & 240 \\ 220 & 240 & 260 & 280 & 300 \\ 264 & 288 & 312 & 336 & 360 \end{matrix}$

The following example shows reading of a single cell from a multi-dimensional field. b is the value of the cell in the second row and the third column of a . The apostrophe is the delimiter.

Example: $b = a [2'3]$

1.16 Define a Function

In *RedCrab* you can define your own functions. The function definition begins with the name on the left, like a definition of a variable. The function symbol and the formal parameter list are in the middle, and the expression is on the right. To get the function symbol, press the keys **Ctrl + 5**.

Example:

$$P = f(x, y) = \sqrt{x^2 + y^2}$$

The example below shows how to call a function that returns the result of an expression. A call of a self-defined function must be marked with the function symbol left of the function name.

$$P = f(x, y) = \sqrt{x^2 + y^2}$$

$$fP(3, 4) = 5$$

$$fP(a, b) = 10$$

$$a = 6 \quad b = 8$$

The arguments can be values, variable names, another function or any expressions.

$$P = f(x, y) = \sqrt{x^2 + y^2}$$

$$fP\left(\frac{144}{a*4}, ft(4)\right) = 10$$

$$a = 6 \quad t = f(x) = 2 * x$$

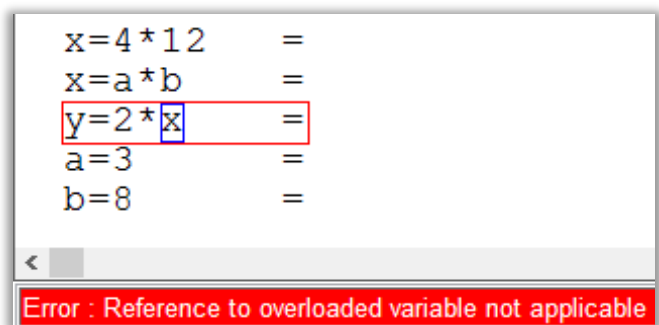
1.17 Scope of Function Parameters

The variables, defined as formal parameter, have own scope inside the function. They can be referenced in the function only and not outside their function. It is allowed and makes no difference, if the same names in the argument list are defined and used elsewhere in the worksheet.

Inside the function you can use in addition to the parameter all other variables, which are defined elsewhere in the worksheet.

1.18 Calculate Selected Formulas

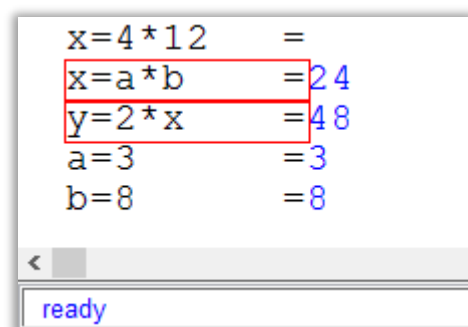
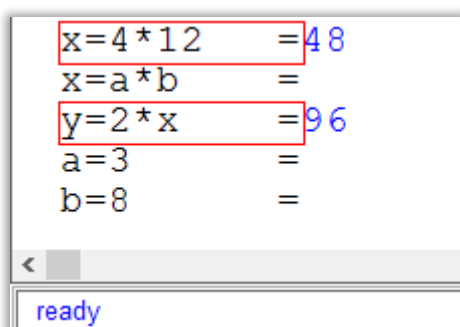
On worksheets which contain a collection of formulas, you can select one or more of them. The following calculation considers the selected formulas only. This can be useful when a worksheet contains different formulas for the same result.



Select the formulas with a click of the right mouse button. The selected formulas are marked with a red frame.

The example on the right shows an error message because the reference variable x has two different definitions.

In the examples below, the variable y is calculated with one of the values of x , controlled by the selection.



The selection is available for the imminent calculation and will be reset when the

calculation terminates.

Result boxes work with selected formulas as well. When in the example above the variable x has a reference to a result box, the result box displays the result of the selected formula.

When a slider is inserted in the worksheet, the slider works only imminently after a calculation which was executed with the **Enter** key or button. After a change on the worksheet, the selection is invalid.

Tutor video: http://www.redchillicrab.com/en/redcrab/tutor/selected_range.html

1.19 Units of Measurement

A feature of RedCrab Math is the ability to calculate with units of measurement. Any numbers can be allocated to a unit. RedCrab has a number of predefined units, which are allocated in groups.

Units of a group and the same dimension can be added and subtracted. Multiplication and division is unrestricted, as long as a meaningful result is calculated.

Not meaningful is hectares * hectares or if a dimension <1 is to be calculated (3km / 2km). Correct is 3 km / 2 = 1.5 km.

Example:

$3\text{km} + 2\text{km} = 5\text{km}$	(kilometre + kilometre)
$3\text{km} + 245\text{m} = 3245\text{m}$	(kilometre + metre)
$12\text{m} + 5\text{yd} = 18.123\text{yd}$	(metre + yard)
$5\text{yd} + 12\text{m} = 16.572\text{m}$	(metre + yard)
$4\text{m} * 5\text{m} = 20\text{m}^2$	(metre * metre)
$2\text{ha} + 950\text{m}^2 = 20950\text{m}^2$	(hectare + square metre)
$650\text{km} / 5.5\text{h} = 118.18\text{km/h}$	(kilometre / hour)

The result is displayed in the unit of measurement of the right operand. In the unit input box on the math menu band, you can input a preferred unit of measurement, which will be shown instead. The preferred unit of measure is ignored if the result is incompatible.

New units can be derived from the predefined units of measurement.

Example: $\text{dm} = 0.1\text{m}$
 $3\text{dm} + 25\text{cm} = 55\text{cm}$

The names of the units can be overloaded by assigning a value to them.

Example: $\text{m} = 15$

In the example above, m is defined as a normal variable which represents a value of 15. In this case the name m cannot be used as a unit of measurement.

1.20 List of Units of Measurement

Group Dimensions

Length

μm	Micrometre	0.000001
mm	Millimetre	0.001
cm	Centimetre	0.01
m	Metre	1
km	Kilometre	1000
mil	Thou	0.0000254
in	Inches	0.0254
ft	Feet	0.3048
yd	Yards	0.9144
ftm	Fathom	1.8288
mi	Miles	1609.344
nmi	Nautical mile	1852
au	Astronomical unit	149598550000

Area

ac	Acres	4046.8564224
ha	Hectares	10000

Volume

L	Litre	0.001
Impgal	ImperialGallon	0.00454609
USliqgal	USLiquidGallon	0.003785411784
USdrygal	USDryGallon	0.00440488377086

Group Weight

mg	Milligram	0.001
g	Gram	1
kg	Kilogram	1000
t	Tonne	1000000
kt	Kilotonne	1000000000
Mt	Megatonne	1000000000000
Gt	Gigatonne	1000000000000000
oz	Ounce	28.349523
lb	Pound	453.59237
tnsh	Short tonne	907184.74
tnlts	Long tonne	1016046.909

Group Temperature

K	Kelvin	-273.15
---	--------	---------

Group Pressure

Bar	Bar	100000
Pa	Pascal	1
kPa	Kilopascal	1000
mmHg	Millimetre of Mercury	133.322387415
atm	Atmospheres	101325
psi	Pound Per Square Inch,	6894.757

Group Energie

J	Joules,	1
kJ	Kilojoules,	1000
cal	Calories	4.1868

kcal	Kilocalories	4186.8
BTU	British thermal unit	1055.056
eV	Electron Volts	$1.60217653 \times 10^{-19}$

Group Power

W	Watts	1
kW	Kilowatts	1000
hp	Horse Power	745.699872
PS	Pferde Staerke	735.49875

Group Time

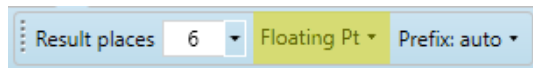
ps	Pico Second	0.000000000001
ns	Nano Second	0.000000001
μs	Micro Second	0.000001
ms	Milli Second	0.001
s	Second	1
h	Hour	3600
d	Day	86400

Groupe Force

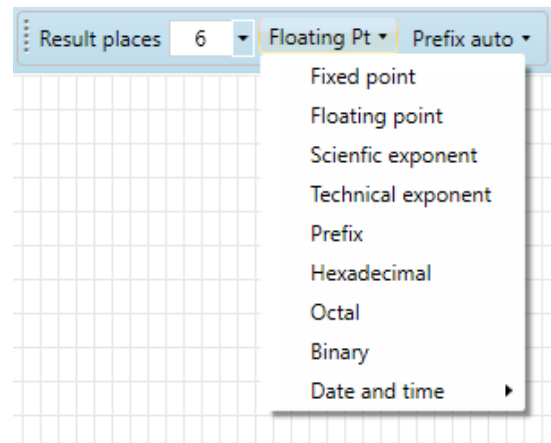
N	Newton	1
lbf	Pound Force	4.4482216152606

2.0 Result Formatting

The tool bar **Result** contains the settings for the result format.



The first menu button on the top row sets the display format. You can choose between fix point, floating point, scientific exponent, technical exponent, prefix, hexadecimal, octal, binary or date time.



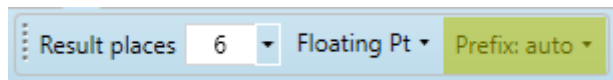
2.1 Result Mode Prefix

In **Prefix** mode the math boxes used SI prefixes instead of exponents. For example, an electrical current of 0.001ampere, or 10^{-3} of an ampere, is written by using the SI-prefix **m** (mill) as 1 mill ampere or 1mA. The SI prefixes are standardized by the International Bureau of Weights and Measures (IBWM).

The list below shows the prefixes which RedCrab used.

Y	Yotta	10^{24}	1.000.000.000.000.000.000.000.000	Quadrillion
Z	Zetta	10^{21}	1.000.000.000.000.000.000.000.000	Trilliarde
E	Exa	10^{18}	1.000.000.000.000.000.000.000.000	Trillion
P	Peta	10^{15}	1.000.000.000.000.000.000.000.000	Billiarde
T	Tera	10^{12}	1.000.000.000.000.000.000.000.000	Billion
G	Giga	10^9	1.000.000.000.000.000.000.000.000	Milliarde
M	Mega	10^6	1.000.000.000.000.000.000.000.000	Million
k	Kilo	10^3	1000	Tausend
x	-	-	1	Eins
m	Milli	10^{-3}	0,001	Tausendstel
μ	Mikro	10^{-6}	0,000.001	Millionstel
n	Nano	10^{-9}	0,000.000.001	Milliardstel
p	Piko	10^{-12}	0,000.000.000.001	Billionstel
f	Femto	10^{-15}	0,000.000.000.000.001	Billiardstel
a	Atto	10^{-18}	0,000.000.000.000.000.001	Trillionstel
z	Zepto	10^{-21}	0,000.000.000.000.000.000.001	Trilliarstel
y	Yokto	10^{-24}	0,000.000.000.000.000.000.000.001	Quadrillionstel

2.2 Specification of a Prefix



If the result of an expression is the distance between two points, the control symbols, #m' displays the result in meters (m).

Examples:

Result: 365	Display: 365m
Result: 3600	Display: 3.6km
Result: 3650000	Display: 3.65Gm

The displayed result: 3.65Gm (Giga meter) is correct, but unusual. Therefore, in **RedCrab** you can preset certain prefixes with the **Prefix** menu. For example, if you choose the prefix k (kilo) the result is displayed as below.

Examples:

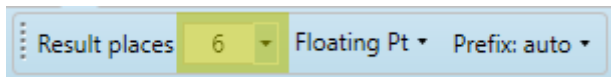
Result: 365	Display: 0.365km
Result: 3600	Display: 3.6km
Result: 3650000	Display: 3650km

RedCrab also has the option to select a group of prefixes or to determine an upper or lower limit. To do this, press the **Ctrl** key and select the lower limit in the **Prefix** menu. Then hold the **Ctrl** key and select the upper limit in the **Prefix** menu. The example below shows results with the limits m (mill) and k (kilo).

Example:

Result: 3650000	Display: 3650 km
Result: 36500	Display: 36.5 km
Result: 365	Display: 365 m
Result: 3.65	Display: 3.65 m
Result: 0.0365	Display: 36.5 mm
Result: 0.000365	Display: 0.365 mm

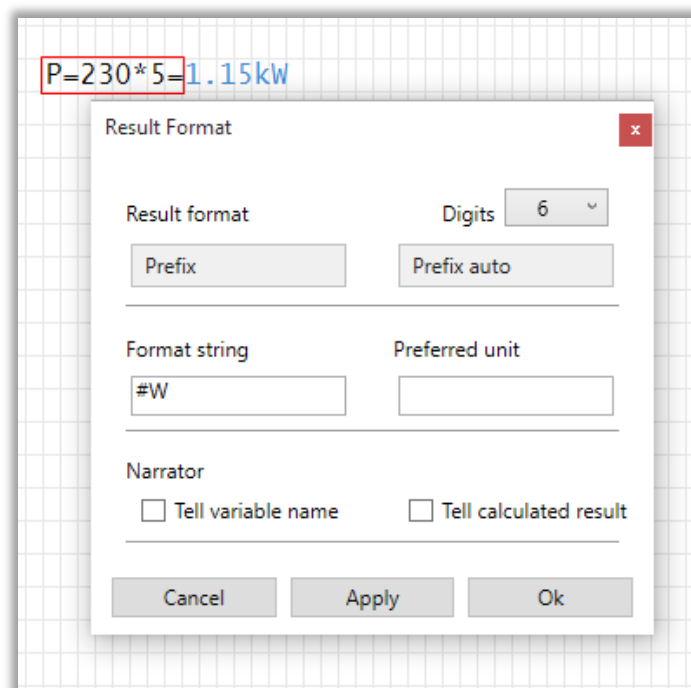
2.3 Number of Decimal Places



The combo box **Result places** determines the number of decimal places to display. In fixed-point mode the combo box sets the number of decimal places to the right of the point. When floating point values are displayed, the setting determines the number of digits without exponents.

2.4 Formatting of Results

The toolbox result format settings are identical for all results. **RedCrab^{PLUS}** can format the results individually for any formula. For this, open the dialog box with double click on the formula symbol (The **P** in the example below).



The menu button on the top sets the main format like the worksheet formatting. The second button changed the prefix, if the main format **Prefix** is selected. The combo box next to the left changed the decimal places.

In the edit line **Format** you can input extended result symbols or text. In the example above (#W) the rhombus (#) is the space for the result, the W represent the unit Watt. The chosen main format is **Prefix**. The display shows: 1.15kW.

Text extentions can be left or right of the rhombus. The following table shows few examples in main mode **Prefix**.

Example:

Result	Format text	Result format
0.012		12m
0.012	#W	12mW
0.012	Power: # W	Power: 12 mW
125	US\$ #	US\$ 125

The format is bounded to the formula symbol.

2.5 Display of Units of Measurement

If you use units of measurement in your calculation, the result is displayed in the unit that was entered on the right in the expression.

Example:

$$2\text{km} + 2\text{mi} = 3.24\text{mi}$$

$$2\text{mi} + 2\text{km} = 5.22\text{km}$$

In the editor line **Unit** you can specify a different unit of measurement that is always displayed when this unit is compatible with the result. The specification is ignored by incompatible results.

2.6 Result speech (Narrator)

With the **Narrator** check boxes you can switch the announcement of the results on or off. The narrator can announce the variable name and the result, or the result only. The correct announcement is dependent on the setting of the **Narrator settings** under the **Tools** menu.

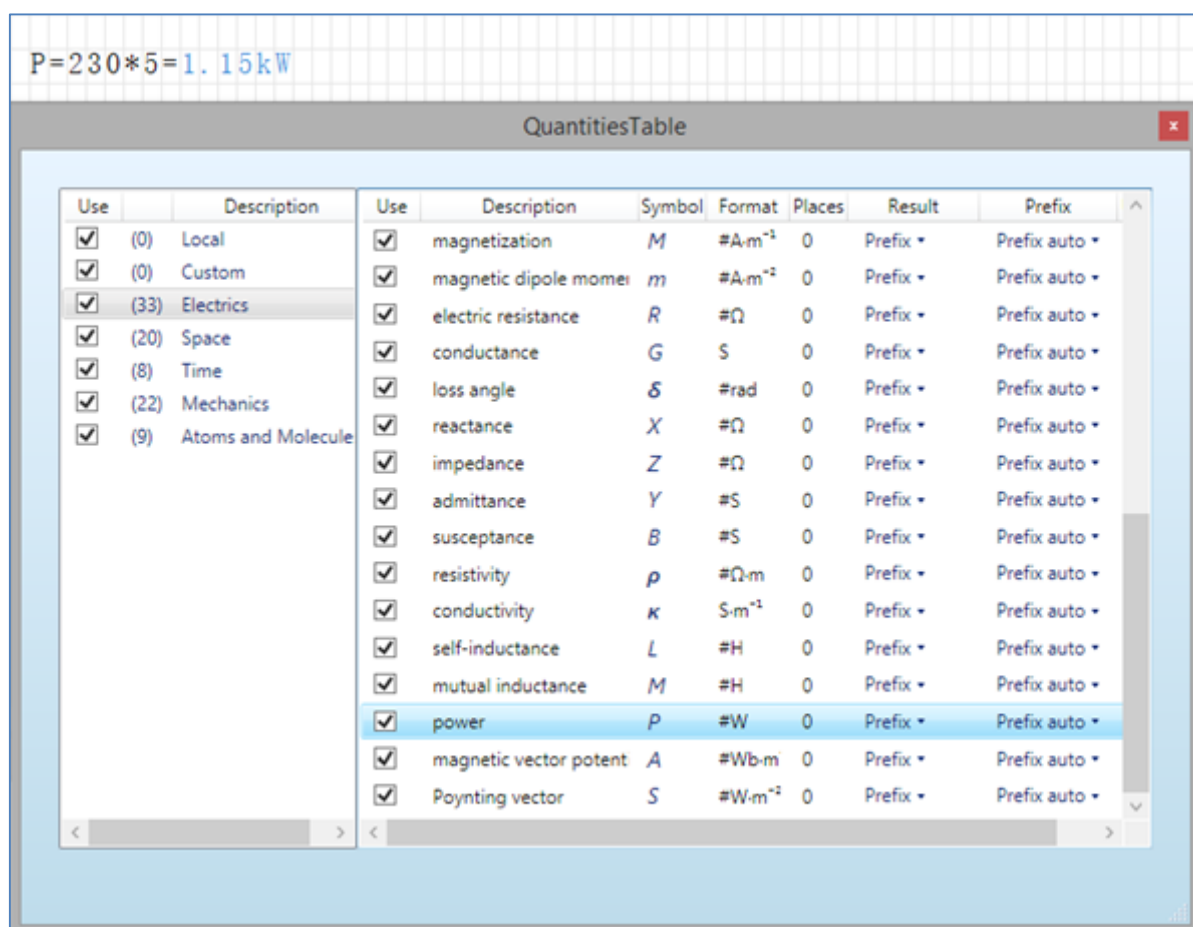
2.7 Formatting Tables

RedCrab contains a number of predefined formattings of the commonly used symbols. Use the menu / button **Formatting table** to open a window which displays the tables of all formattings.

The table **Local** contains formattings, defined by yourself in this worksheet. These formats are saved with the document and are only in this document available.

In the **Custom** table you can compose a personal list of defined formats, by copying them from the **Local** or the predefined tables. The **Custom** table is saved with the default settings of the programme and is available on each worksheet.

The formats can be individually or in groups enabled or disabled with check boxes.



2.8 Edit Formatting

A click with the right mouse button on the formula symbol opens a popup menu. With this you can delete an entry or copy them to the **Custom** table. You cannot delete the predefined table entries permanently; they are available again after the next restart.

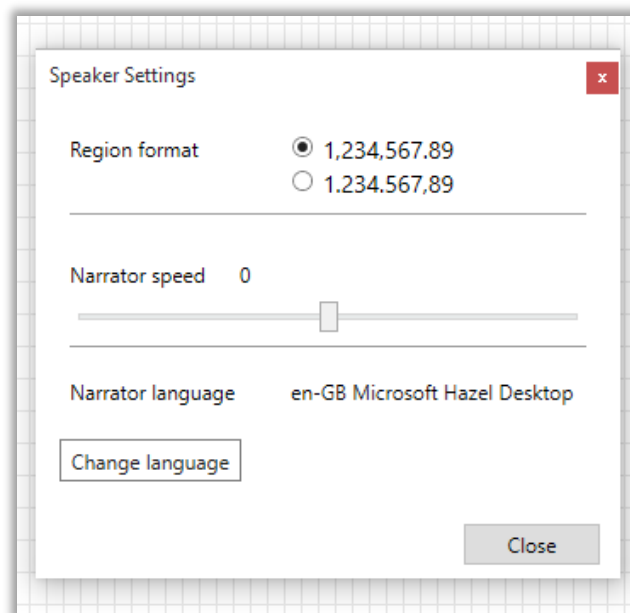
The column **description**, **format** and **places** can be edited. Double-click on the text that you want to change to open the editor. Terminate the input with **Enter**.

Result and Prefix can be changed using pull down menus.

The formats are saved as *.xml files in the subdirectory/tools/symbols by the RedCrab start up directory.

2.9 Narrator settings

Narrator settings in the menu **Tools** opens a dialog box to change the narrator settings.



It is important that the **Region format** choice is correct, to avoid wrong result. This setting determines whether a point or a comma is used as the decimal separator. The format depends on the language used.

Under **Narrator speed**, you can change the speed of the speech.

Narrator language displays the used language. If multiple languages are installed on your system, the language can be set with the menu *Change language* to another language.

2.10 Display Tables

Like all results, tables are displayed to the right of the equal sign. If necessary, the worksheet will automatically enlarge. If the required space on the worksheet is not empty, the table in a separate box is displayed which is placed on the worksheet.

The separate Box has a slightly offset and a border is displayed when you click on the table. The separate box has a popup menu which you can use to display the table in a separate scrollable window.

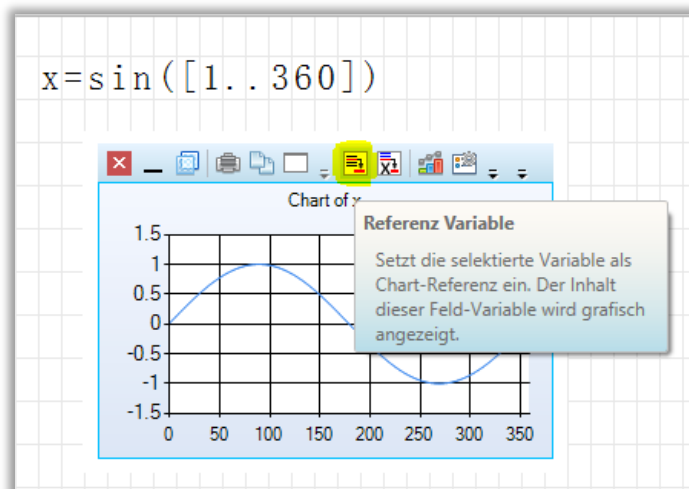
For large tables, the display in a separate box with a new window is preferred. You can force the display in a separate box by writing a colon following the equal sign.

3.0 Display Results Graphically with Charts



RedCrab provides chart boxes to display results graphically. To open a chart box, select a range on the worksheet with the mouse pointer and click the *Insert Chart box* button. Later you can change the size and the position of the box with the mouse pointer.

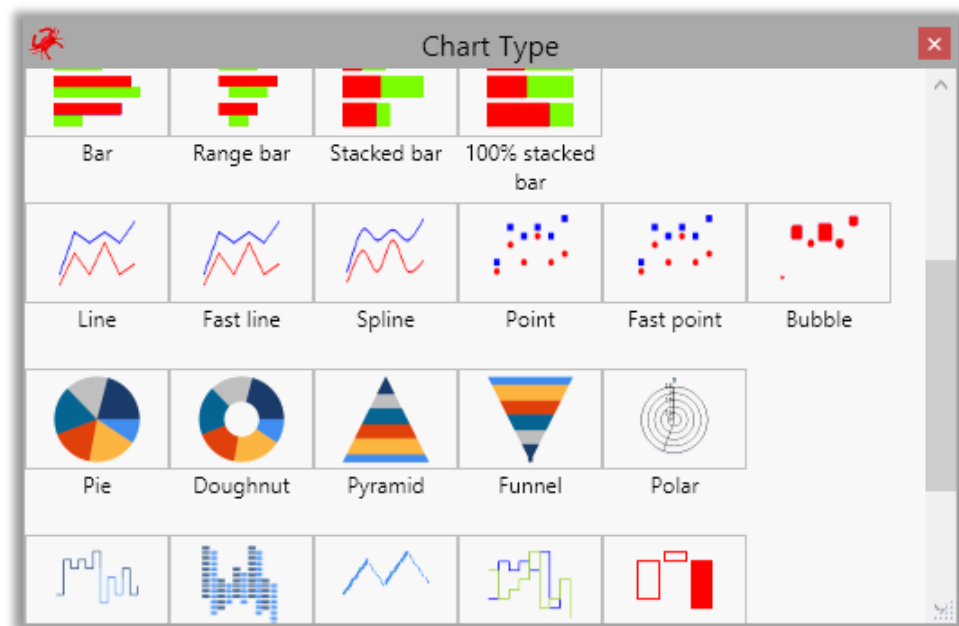
To make a reference to a variable, position the cursor per mouse click on the variable; then click the button *Reference variable* or click the item on the chart popup menu.



3.1 Chart Type

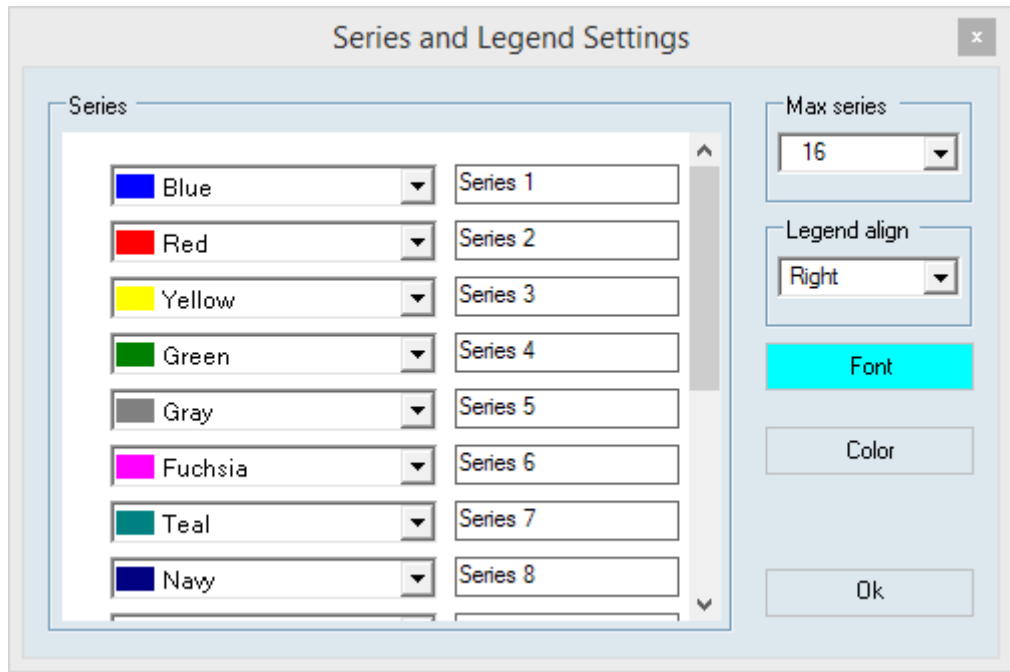


The button ***Chart types*** opens a dialog box to choose between different chart types. To choose a chart type, first click on the chart box. Then click the chart icon. Do note that different chart types need different data format. All icons show tool tips with notes if the mouse is moved over them.



3.2 Legend Settings

Legend settings open a dialog window to change the series names, colours and the legend position.



In **RedCrab** 16 different colours for drawing of series are presented. If you use more than 16 series in a chart, beginning with series 17, the colours will repeat. The series name is displayed with the word **Series** and a current number.

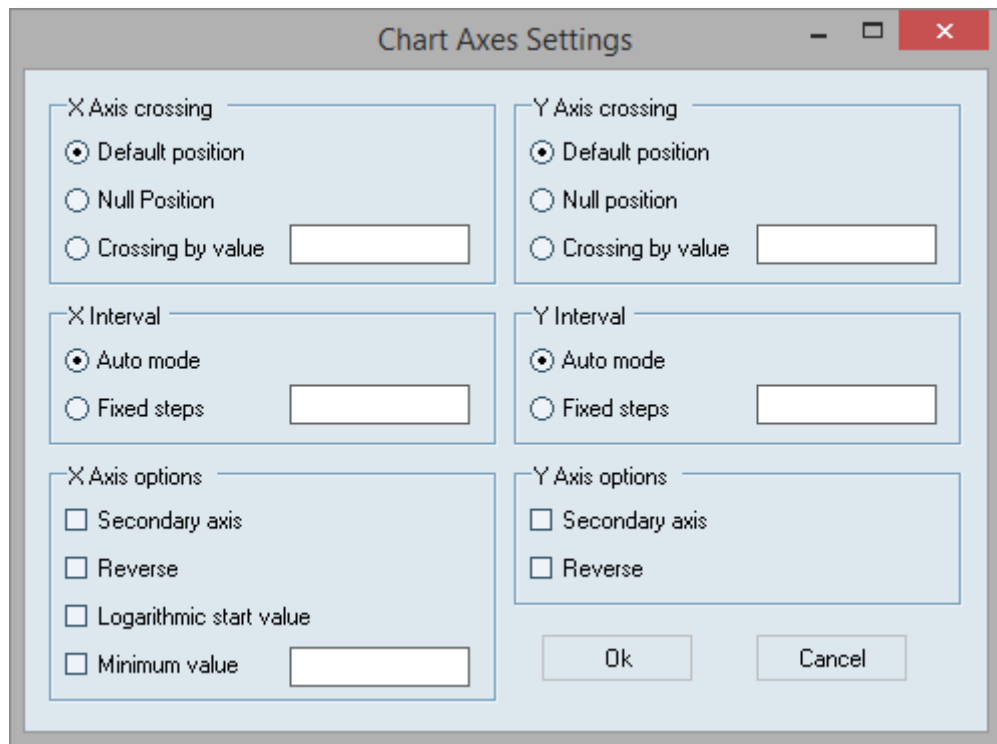
The dialog window contains 16 combo boxes to change the series colours and 16 editor boxes to assign the series name. If you need more than 16 series you can extend the list with the combo box **Max series**.

Max series does not limited the number of series of a chart box, it only specify the length of the list. If **Max series** is set to 16, and you use 20 series in the chart box, **RedCrab** uses for series 1 to 16, the dialog box colours and text. The series 17 to 20 uses the preset colour and the name **Series** with a current number.

Legend align sets the legend position.

3.3 Axis Settings

Axis settings open a dialog window for input of the axes properties. The picture below shows the Axis dialog window.

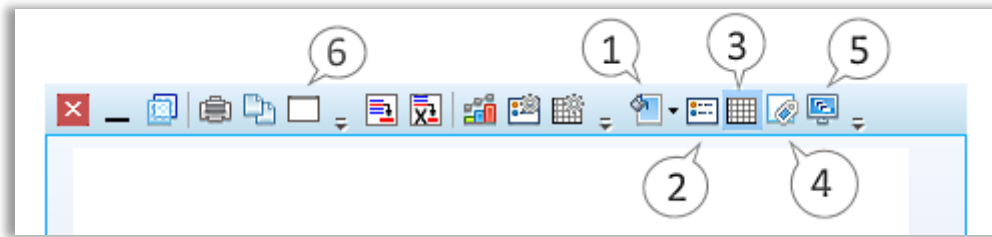


Axis crossing sets the axis position. By default setting, the Y axis is always left from the chart and the X axis at the bottom of the chart. Alternate you can position the axis on the null value of the scale or to a manual entered value.

Interval sets the scale steps on automatic or a manual entered value

Axis options displays with **Secondary axis** an additional axis on the top or on the right side. **Reverse** invert the axes scales. The values of the **X** axis increase from right to left instead from left to right. The values of the **Y** axis increase from top to bottom instead from bottom to top.

3.4 Chart Options



1) **Background** selects the chart box background.

- **Flat** displays the chart box with a white background without frame.
- **Single border** displays the chart box with a white background and a small frame.
- **Color** displays the chart box with a single-colored background and a small frame.
- **Default** displays the chart box with default color background and a small frame.

2) **Show legend** switches the legend to the state show or hide.

3) **Show axis** switches the axis in show or hide mode.

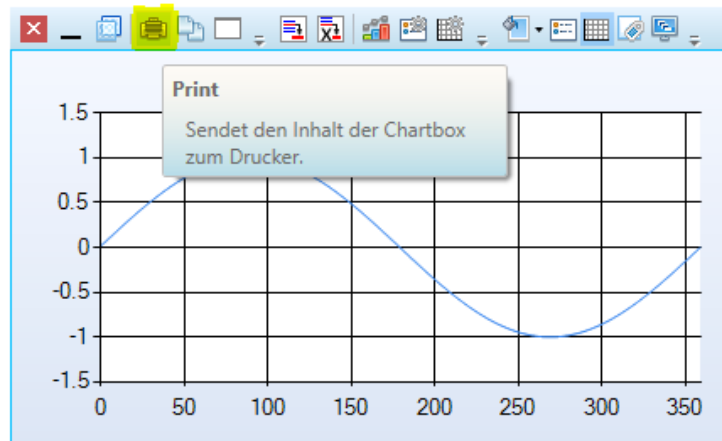
4) **Show labels** shows labels with decimal values.

5) **3D chart area** displays the chart area in 3D-design.

6) **Undocked** displays the chart box in a separate window.

3.5 Print Chart box

With the Print button, you can print the chart box content. A mouse click on the button opens the printer dialog box.



4.1 Text Box

RedCrab provides inserting of text into your worksheet. To create a text box, first select a range on the worksheet and click the button **New text box** on the **Insert** tool bar. The procedure is identical to the creation chart box.



If you want to load a text file to your worksheet, click the ***Open text file*** button. ***RedCrab*** create automatically a new text box to display the file.



4.2 Insert Pictures

For complex technical calculations, it might be useful to include pictures to mathematical formulas. **RedCrab** supports insert of images in any position on the worksheet.

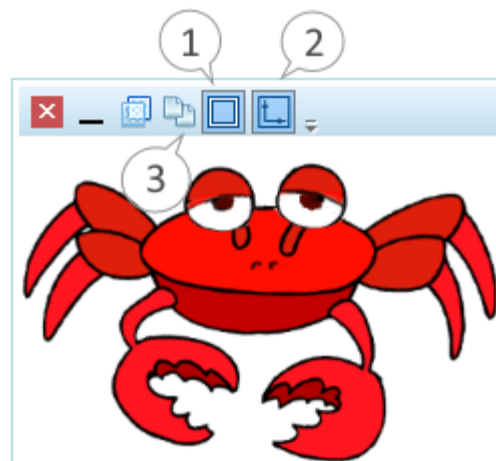
Click the button **Image from clipboard** on the **Insert** tool box to paste an image from the clipboard.



Alternate you can load image files into the worksheet. To do this, open the image file browser with a click on **Import image file** button. Then select the image file. **RedCrab** can import the image formats Windows Bitmap (*. *bmp*) *.*jpg*, *.*gif*, *.*png* and *.*tif*.



You change the image position or resize by dragging the image border.



The button **Aspect ratio** (2) locks the image width to height ratio. The button **Original size** (1) resets the size to the original.

The button **Copy** (3) copies the picture to the clipboard.

4.3 Programm Box



The button **Program box** on the **Insert** tool bar opens an editor box for programming of your own functions.

RedCrab supports programming of functions with its own program language.

Worksheet formulas have access to all functions of **RedCrab** programs. From **RedCrab** program, you can call all functions in other **RedCrab** program modules.

The command language is easy to learn, especially for users without programming experience. The syntax of the interpreter is an extension of the worksheets syntax. That means, all the mathematical functions of the worksheet are also available. Likewise, the definition of variables and data fields is identical with the worksheet.

In addition, the editor contains commands for programming functions, conditional branching (**If**, **Elseif**, **Else**) and loops (**While**).

For more information read the **Programmers Manual**.

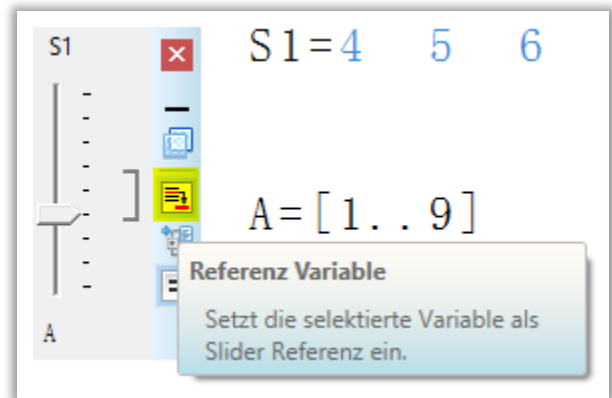
4.4 Insert Slider

A slider can be used instead of a variable in a formula or as a parameter of a function. The slider returns the individual value of a series of numbers depending on the position of the slider button. The sequence of numbers is defined as a field variable.



To add the reference to the variable, place the cursor on the variable; then click the button **Referenz variable**. The name of the reference variable is displayed at the bottom in the slider

The name of the sliders is used in the formula instead of a variable, in the example above S1.



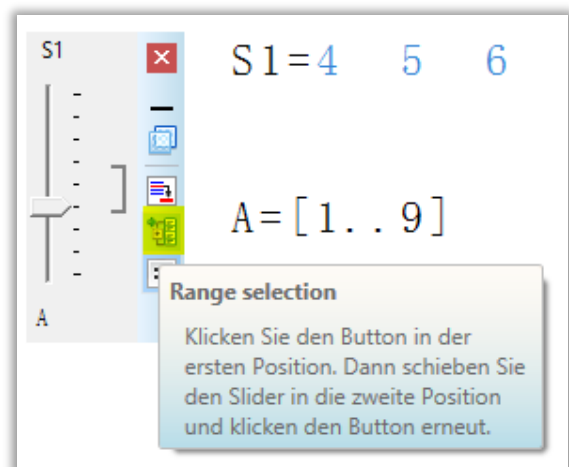
The name of the slider is automatically assigned. You can change the name by clicking on the name. A text box appears where you can change the name.

If the Autocalc function activated, a new calculation starts when the slider is moved.

4.5 Range Selection

The slider provides the selection of a data range. Instead of single value, the slider output value is a data field, which contains the values of the selected range.

To select a range, first mark the actual position with a click on **Range selection**. Then move the slider button to the second position and repeat **Range selection**. The image shows an example with the selected values 4 and 6.



4.6 Insert Label

RedCrab supports the creation and formatting of labels. Labels can be positioned in chart and image boxes. They can display text or results of calculations. Labels can display single values, but not data fields.



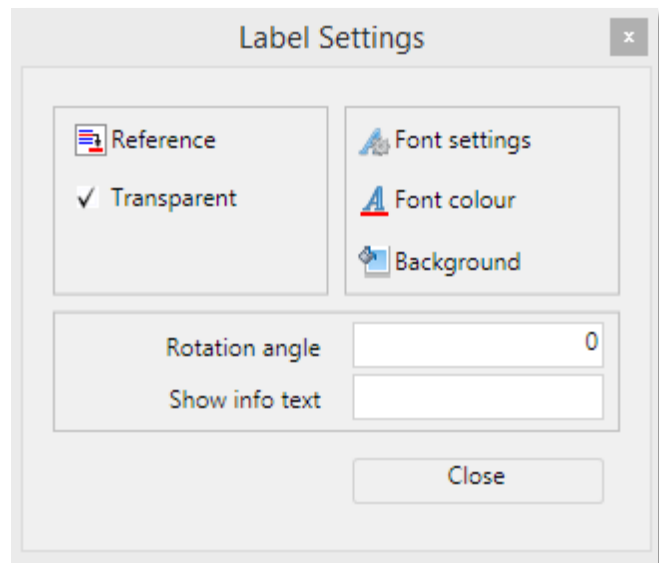
To create a label, first choose the target chart or image box with a mouse click. Then click the button **New Label**. Finally drag the label with the mouse to the desired position.

Double click the label to open a dialog window to enter label settings.

If you want to display a text label, write the text in the editor row next to **Show info text**. Terminate the text with **Enter**. The size of the label automatically adapts to the content.

If the label is to display the result of a calculation, you must connect the desired variable to the label.

Place the cursor on the reference variable. Then click the **Reference** button.



Additional settings

Transparent displays the background transparent

Background opens a dialog box to choose the background colour

Font settings open a dialog box for font settings

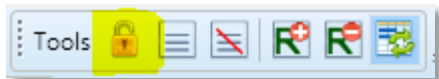
Font colour opens a dialog box to choose the font colour

Rotate rotates the label. Input the rotation in degree to the editor box beside **Rotation angle**.

5.0 Toolbar

5.1 Page Lock

Page Lock blocks the editor's page for additional entries. This function protects unintentional changes made. For data input, cells can be unlocked with **Unlock Cell**.



5.2 Cell Unlock

The **Cell unlock** button unlocks cells in a locked page for data entry. Select the cells by mouse, and then click **Unlock Cell**. The unlocked fields are marked with an underscore.

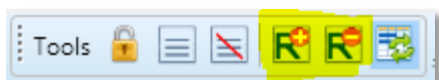
Reset the remark function above with the **Reset Cell** button by using the same step



5.3 Remark

With the **Set remark** (R+) button mark selected text in the worksheet as a comment. This function can be performed with the function key F2. Comments are ignored by the calculator. The selected data is displayed in green.

The button **Clear remark** (R-) undoes the remark function above by using the same step.



5.4 Autocalc

If ***Autocalc*** option is set, the calculator start a new calculation if you enter an equal symbol

5.5 Tooltip Language

The Item ***Tooltip language*** in the menu ***Tools*** chooses the tool tips language. The language English and German is in ***RedCrab*** installed. Additional languages can be added with language files.

5.6 Keyboard Settings

With the Item ***Keyboard settings*** in the menu ***Tools*** you can select the keyboard.

The keyboard inputs in this description refer to an English keyboard in the country's setting ***English-US***. When using another keyboard or regional setting, some functions are acquired with other key combinations.

Attached you will find images about key codes of the alternative keyboards and the occupancy of the ***Ctrl*** functions.

5.7 View Menu (Toolbar)

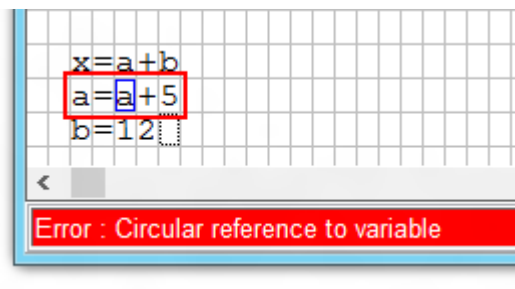
Function panel	displays a window that contains the implemented functions.
Number pad	opens a virtual number pad on the display.
Symbol pad	displays a window with a symbol pad.
Virtual keyboard	opens a virtual keyboard.
Formatting table	opens a window with all pre-defined result formats.
Show grid	draws a grid on the worksheet.
Show border	draws a border line on all boxes.

Toolbox size changes the size of the toolbox buttons.

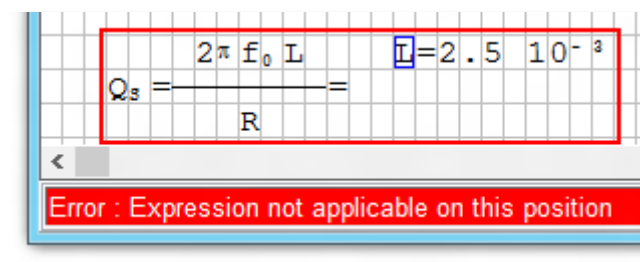
Show boxes toolbars change the boxes toolbars visibility visible or hide.

5.8 Error Messages

For error location RedCrab marks the cell in where an error is detected with a blue frame. It also marks the incorrect formula with a red frame.



The marking of the entire formula simplifies the localization of errors that cause a false positioning. In the example below, an invalid assignment is signaled. The red selected box indicates, however, that two formulas were joined because the distance is too close. In this example the adjustment of the distance (column space) is 4 columns; the distance between the formulas is only 2 columns.



5.9 License Activation

This menu opens a dialog box to update the duration of the shareware. To do this, you must insert your registered email and registered key. This update is necessary

if you have acquired an extension of the duration through the purchase of a license or participate in a promotion.

The acquired duration is stored in *RedCrab* setup. If you lost the information by reinstall of the operating system or by deleting the configuration data, you can reactivate it again.

For activating, an online connection is required.

6.0 Functions and Operators

The following section describes the RedCrab Math functions and text operators. All the functions can be entered per mouse click on the function panel or via the keyboard. The function panel includes tool tips with short description and examples for all functions.

The panel's button size can be changing with the mouse wheel.

6.1 Standard Functions

Abs *Abs* returns the absolute value of numbers and fields.

Example: $x = \text{Abs}(y)$

$$\begin{aligned} X &= \text{abs}(4.56) = 4.56 \\ X &= \text{abs}(-4.56) = 4.56 \end{aligned}$$

Ceil Returns the smallest integer that is not less than the argument.

Example: $\text{ceil}(-2.3) = -1$
 $\text{ceil}(2.5) = 3$

DTime The function ***DTime*** returns the ***DateTime*** value of the given year, month, day, hour, minute and second. The argument must be a data field that includes six cells which contains the value of year, month, day, hour, minute and second.

The year must be between 1 and 9999.

Valid Month values are 1 through 12.

Valid Hour values are 0 through 23.

Valid Min and Sec values are 0 through 59.

Valid Day values are 1 through 28, 29, 30, or 31, depending on the Month value. For example, the possible Day values for month 2 (February) are 1 through 28 or 1 through 29, depending on whether or not the Year value specifies a leap year.

Example: `d = dtime([Y, M, D, h, m, s])`

A call of ***DTime*** with the argument ***0*** returns the current date and time.

Example: `current = dtime(0)`

DTimef The function ***DTimef*** returns a data field that includes six cells which contains the value of year, month, day, hour, minute and second of the arguments ***DateTime*** value.

Example `dtimef(d) = 2012 4 12 14 27 18`

Floor Returns the largest integer that is not greater than the argument.

Example: `floor(-2.3) = -3`
`floor(2.5) = 2`

Frac ***Frac*** returns the fractional part of an argument.

Example: `x = frac(y)`

$$X = \text{frac}(4.67) = 0.67$$

Int ***Int*** returns the integer part of a value; that is, the value rounded toward zero.

Example: `x = int(y)`
 `X = int(4.67) = 4`

Rnd ***Rnd*** returns a random integer number within the range $0 \leq X \leq \text{Range}$.

Example: `x = rnd(y)`

Round ***Round*** returns a value rounded to the nearest whole number.

Example: `x = Round(y)`

`round(2.6) = 3`
`round(3.5) = 4`
`round(2.5) = 2`

If *y* is exactly halfway between two whole numbers, the result is always the even number. This method of rounding is often called "Banker's rounding".

Sign Returns a value indicating the sign of a number.

1: value is greater than zero.
0: value is equal to zero.
-1: value is less than zero.

Sqr The ***Sqr*** function returns the square of the argument.
Example: `sqr(4) = 16`

Sqrt The result of ***Sqrt*** is the square root of the argument.
Example: `sqrt(4) = 2`

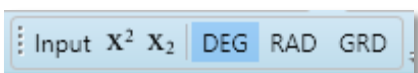
URnd ***URnd*** fills a field with a series of random numbers between 0 and the highest argument of the field. In contrast to ***Rnd***, which also can be used for fields, ***URnd*** returns a set of unique numbers.

Example: `a = urnd([1..5, 45])`
`B = urnd([44..45])`

Both examples return a list of six different numbers between one and 45.

6.2 Scientific Functions

The buttons ***Degree***, ***Radian*** and ***Gradient*** in the ***Input*** toolbar determine whether the parameters of trigonometric functions be specified in degrees, radian or grads.



ACos inverse cosine

ASin inverse sine

ATan inverse tangent

Cos cosine

Cosh hyperbolic cosine

<i>Cot</i>	cotangent
<i>Deg</i>	converts radian in degrees
<i>Exp</i>	exponent to Euler's constant: 2.7182818284590452...
<i>Ln</i>	natural logarithms to base e (2.7182818284590452...)
<i>Log</i>	logarithms base 10
<i>Log2</i>	logarithms base 2
<i>Log8</i>	logarithms base 8
<i>Log16</i>	logarithms base 16
<i>Rad</i>	convert degrees in radians
<i>Sin</i>	sine
<i>Sinh</i>	hyperbolic sine
<i>Tan</i>	tangent
<i>Tanh</i>	hyperbolic tangent

Alternated notations (enter only with keyboard)

<i>Ld</i>	logarithms base 2 (equal to log2)
<i>Lg</i>	logarithms base 10 (equal to log)
<i>Log10</i>	logarithms base 10 (equal to log)

6.3 Programmer Functions and Operators

And The logical ***And*** operator performs bitwise AND manipulation on integer operands

Example: $Z = X \text{ and } Y$

Div The ***Div*** operator returns the result of an integer number division without remainder. If floating point numbers are entered, the ***Div*** operator cuts off all digits after the decimal point before executing the division ***Div***.

Example: $11 \text{ div } 3 = 3$
 $11.2 \text{ div } 3.9 = 3$

Excl clears the bit from the first argument, which is determined in the second argument.

Example: $Z = \text{excl}(X, Y)$

In the example above ***Excl*** clears the bit number ***Y*** in argument ***X***

Example: $\text{excl}(15, 4) = 7$

Incl sets the bit from the first argument, which is determined in the second argument.

Example: $Z = \text{incl}(X, Y)$

In the example above ***Incl*** sets the bit number ***Y*** in argument ***X***

Example: $\text{incl}(8, 3) = 12$

Mod The ***Mod*** operator returns the remainder of the division of two integer numbers. If floating point numbers are entered, the ***Mod*** operator cuts off all digits after the decimal point before executing the division ***Mod***.

Example: $11 \text{ mod } 3 = 2$
 $11.7 \text{ mod } 3.9 = 2$

Not The logical ***Not*** function performs bitwise negation on integer operands.

Example: $Z = \text{not}(X)$

Or The logical ***Or*** operator performs bitwise OR manipulation on integer operands.

Example: $Z = X \text{ or } Y$

Shl performs an arithmetic left shift on a bit pattern. The value of ***Y*** is interpreted modulo 32. Thus for example, if ***X*** is 40, ***X*** is interpreted as **8**.

Examples: $Z = \text{shl}(X, Y)$
 $\text{shl}(9, 2) = 36$

Shr performs an arithmetic right shift on a bit pattern. The value of ***Y*** is interpreted modulo 32. Thus for example, if ***X*** is 40, ***X*** is interpreted as **8**.

Examples: $Z = \text{shr}(X, Y)$
 $\text{shr}(8, 2) = 2$

Xor The logical ***Xor*** operator performs bitwise XOR manipulation on integer operands.

Exemple: `Z = X xor Y`

6.4 Data Fields Functions

Aver The function ***Aver*** returns the mean values of successive elements of fields. The result is always one element smaller than the original field.

Example: `a = [1..5]^2 = 1 4 9 16 25`
`b = aver(a) = 2.5 6.5 12.5 20.5`

Cols The function ***Cols*** returns the number of columns of a two dimensional data field.

Example: `x = [1..4;12..15]`
`c = cols(x) = 4`

Count Return the number of elements of one- or multidimensional fields.

Example: `z= count(x)`
`x= [9, 7, 2, 8, 12, 3, 5]`
`count(x) = 7`

Diff Calculates the difference values of successive of a set of numbers

Example: `diff([2, 5, 9, 11]) = 3 4 2`

Dim returns the number of dimensions of a multi dimensional data field.

Example: `x = [1..4; 12..15]`
`dim(x) = 2`

Fill fills the data field of the first argument with the value of the second argument.

Example: `x = fill([1..5], 8) = 8 8 8 8 8`

Join connects 2 one or two-dimensional fields with each other.

Example: `a = [1..5] = 1 2 3 4 5`
`b = [6..10] = 6 7 8 9 10`

`c = join(a, b) = 1 2 3 4 5`
`6 7 8 9 10`

If the fields are different lengths, the shorter field is filled with zeros.

```
x = [11..18] = 11 12 13 14 15 16 17
d = join(x, c) = 11 12 13 14 15 16 17
                  1  2  3  4  5  0  0
                  6  7  8  9 10  0  0
```

Maxi returns the greatest value of the argument list.

Example: `z = maxi(x)`
`X = [9, 7, 2, 8, 12, 3, 5]`
`maxi(x) = 12`

Mini returns the smallest value of the argument list.

Example: `z = mini(x)`
`X = [9, 7, 2, 8, 12, 3, 5]`

`mini(x) = 2`

Patt fills the data field of the first argument with the pattern of the second argument.

Example:

`x = patt([1..10], [1,1,2])=1 1 2 1 1 2 1 1 2 1`

Rows returns the number of rows of a two dimensional data field.

Example: `x = [1..4; 12..15]`

`r = rows(x) = 2`

6.5 Matrix Functions

Det returns the determinant of a 2x2 or 3x3 matrixes. More information of determinants can be found at:

Example: `d = det(A)`

Invx inverse a 2x2 or 3x3 matrix. If the matrix is not invertible, ***RedCrab*** displayed an error message.

Example: `A1 = invx(A)`

Mulx ***Mulx*** is an operator for multiplication of matrices. Multiplication of two matrices with ***Mulx*** is possible only if the number of columns

of the left matrix is the same as the number of rows of the right matrix.

Example:
$$\mathbf{x} = \begin{bmatrix} 1, 2, 3 \\ 4, 5, 6 \\ 7, 8, 9 \end{bmatrix} \text{ mul } \mathbf{x} \begin{bmatrix} 2, 4 \\ 3, 5 \\ 6, 8 \end{bmatrix} \quad \begin{bmatrix} 26 & 38 \\ 59 & 89 \\ 92 & 140 \end{bmatrix}$$

The result is a matrix whose entries are given by dot product of the corresponding row of the left operand and the corresponding column of the right operand:

$$\begin{array}{ll} (1*2 + 2*3 + 3*6) & (1*4 + 2*5 + 3*8) \\ (4*2 + 5*3 + 6*6) & (4*4 + 5*5 + 6*8) \\ (7*2 + 8*3 + 9*6) & (7*4 + 8*5 + 9*8) \end{array}$$

Trans producing the transpose of a matrix A^T , which is computed by swapping columns for rows in the matrix X .

Example:
$$\mathbf{x} = \begin{bmatrix} 1, 2, 3 \\ 4, 5, 6 \\ 7, 8, 9 \end{bmatrix}$$

$$\text{Trans}(\mathbf{x}) = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

6.6 Statistics Functions

CuSum returns the calculation of a cumulative sum of one- dimensional fields.

Example:
$$\mathbf{z} = \text{cusum}(\mathbf{x})$$

$$\text{cusum}([2, 4, 7, 3, 9]) = -3 \quad -4 \quad -2 \quad -4 \quad 0$$

DSort sorts field elements from high to low values (sort descending). Complex fields are sorted based on first row values. For *sort ascending* see ***Sort*** below.

Example: `z = dsort(x)`

LQuart returns the value of the first quartile (lower quartile) of a sorted list. In the following example in a field of 10 elements, the position of the first quartile is $(10 \times \frac{1}{4}) = 2.5$, rounded up to 3.

Example:

`lquart([3, 6, 7, 8, 8, 10, 13, 15, 16, 20]) = 7`

See ***UQuart*** and ***QRan*** below.

Mean returns the mean value of fields. In multidimensional fields the result is the mean of all elements.

Example: `z = mean(x)`

Median returns the median value of fields. In multidimensional fields the result is the median of all elements.

Example: `z = median(x)`

Prod returns the product of all elements of fields.

Example: `z = prod(x)`
`x = [9, 7, 2, 8, 12, 3, 5]`
`prod(x) = 181440`

QRan results the area from the first to 3rd quartiles of a sorted list. The following example shows the result of a field with 10 elements.

Example:

```
qran([3,6,7,8,8,10,13,15,16,20])=7 8 8 10 13 15
```

Sort sorts field elements from low to high values (sort ascending). Complex fields are sorted based on first row values.

For *sort descending* see ***DSort*** above.

Example: `z = sort(x)`

SStDev returns the standard deviation of values in one-dimensional fields. Use ***SStDev*** if the field contains sample data. If the field contains all evaluated data, see ***StDev*** below.

Example: `z = sstdev(x)`

StDev returns the standard deviation of values in one-dimensional fields. Use ***StDev*** if the field contains all evaluated data. If the field contains samples, see ***SStDev*** above.

Example: `z = stdev(x)`

Sum returns the sum of the elements in fields. The function can be called by the Greek letter Σ .

Example: `z = sum(x)`
`x = [9,7,2,8,12,3,5]`
`sum(x) = 46`

SVari returns the variance of values in one-dimensional fields. Use ***SVari*** if the field contains sample data. If the field contains all evaluated data, see ***Vari*** below.

Example: `z = svari(x)`

UQuart returns the value of the third quartile (upper quartile) of a sorted list. In the following example, in a field of 10 elements the position of the third quartile is $(10 \times \frac{3}{4}) = 7.5$, rounded up to 8.

Example: `UQuart([3,6,7,8,8,10,13,15,16,20])=15`

See Lquart and QRan above.

Vari returns the variance of values in one-dimensional fields. Use ***Vari*** if the field contains all evaluated data. For samples see ***SVari*** above.

Example: `z = vari(x)`

6.7 Financial Functions

FDDB returns a value specifying the depreciation of an asset for a specific time period using the double-declining balance method or some other method you specify.

Syntax: `fddb (Cost, Salvage, Life, Period)`

Optional: `fddb (Cost, Salvage, Life, Period, Factor)`

Parameter:

Cost specifying initial cost of the asset.

Salvage specifying value of the asset at the end of its useful life.

- Life*** specifying length of useful life of the asset.
- Period*** specifying period for which asset depreciation is calculated.
- Factor*** specifying rate at which the balance declines. If omitted, 2 (double-declining method) is assumed.

FFV returns the future value of an annuity based on periodic, fixed payments and a fixed interest rate.

Syntax: `ffv (Rate, NPer, Pmt)`

Optional: `ffv (Rate, NPer, Pmt, PV)`
 `ffv (Rate, NPer, Pmt, PV, Due)`

Parameters:

- Rate*** is the interest rate per period. For example, if you get an annual percentage rate (APR) of 6 percent and make monthly payments, the rate per period is 6/12, or 0.5.
- NPer*** specifies the total number of payment periods in the annuity. For example, if you make monthly payments, you have a total of 4×12 (or 48) payment periods.
- Pmt*** specifying payment to be made each period. Payments usually contain principal and interest that doesn't change over the life of the annuity.
- PV*** optional specifying present value (start value) of a series of future payments. If omitted, 0 is assumed.
- Due*** optional specifying when payments are due. This argument must be either 0, if payments are due at the end of the payment period, or 1, if payments are due at the beginning of the period. If omitted, 0 assumed.

FIPmt specifying the interest payment for a given period of an annuity based on periodic fixed payments and a fixed interest rate.

Syntax: `fipmt (Rate, Per, NPer, PV,)`

Optional: `fipmt (Rate, Per, NPer, PV, FV)`
 `fipmt (Rate, Per, NPer, PV, FV, Due)`

Parameters:

Rate is the interest rate per period. For example, if you get a car loan at an annual percentage rate (APR) of 6 percent and make monthly payments, the rate per period is 6/12, or 0.5.

Per specifies the payment period in the range 1 through NPer.

NPer specifies the total number of payment periods in the annuity. For example, if you make monthly payments on a four-year car loan, your loan has a total of 4×12 (or 48) payment periods.

PV is the present value (or lump sum). For example, when you borrow money to buy a car, the loan amount is the present value.

FV optional specifying future value or cash balance you want after you have made the final payment. For example, the future value of a loan is \$0 because that is its value after the final payment. However, if you want to save \$50,000 during 18 years for your child's education, then \$50,000 is the future value. If omitted, 0 is assumed.

Due optional specifies when payments are due. This argument must be either 0, if payments are due at the end of the payment period, or 1, if payments are due at the beginning of the period. If omitted, 0 assumed

FIRR returns the internal rate of return for a series of periodic cash flows (payments and receipts).

Syntax: `firr (ValueArray)`

Optional: `firr (ValueArray, Guess)`

Parameter:

ValueArray is an array of cash flow values. The array must contain at least one negative value (a payment) and one positive value (a receipt).

Guess optional specifying value you estimate will be returned by FIRR. If omitted, Guess is 10 percent

FMIRR returns the modified internal rate of return for a series of periodic cash flows (payments and receipts).

Syntax: `fmirr (ValueArray, FinanceRate, ReinvestRate)`

Parameter:

ValueArray is an array of cash flow values. The array must contain at least one negative value (a payment) and one positive value (a receipt).

FinanceRate specifying interest rate paid as the cost of financing.

ReinvestRate specifying interest rate received on gains from cash reinvestment

FNPer returns the number of periods for an annuity based on periodic fixed payments and a fixed interest rate.

Syntax: `fnper (Rate, Pmt, PV)`

Optional: `fnper (Rate, Pmt, PV, FV)`
 `fnper (Rate, Pmt, PV, FV, Due)`

Parameters:

Rate is the interest rate per period. For example, if you get a car loan at an annual percentage rate (APR) of 6 percent and make monthly payments, the rate per period is 6/12, or 0.5.

Pmt specifying payment to be made each period. Payments usually contain principal and interest that does not change over the life of the annuity.

PV is the present value (or lump sum). For example, when you borrow money to buy a car, the loan amount is the present value.

FV optional specifying future value or cash balance you want after you have made the final payment. For example, the future value of a loan is \$0 because that is its value after the final payment. However, if you want to save \$50,000 during 18 years for your child's education, then \$50,000 is the future value. If omitted, 0 is assumed.

Due optional specifies when payments are due. This argument must be either 0, if payments are due at the end of the payment period, or 1, if payments are due at the beginning of the period. If omitted, 0 assumed

FNPV returns the net present value of an investment based on a series of periodic cash flows (payments and receipts) and a discount rate.

Syntax: `fnpv (Rate, ValueArray)`

Parameter:

Rate the discount rate over the length of the period.

ValueArray is an array of specifying cash flow values. The array must contain at least one negative value (a payment) and one positive value (a receipt)

FPmt specifying the payment for an annuity based on periodic, fixed payments and a fixed interest rate.

Syntax: fpmt (Rate, NPer, PV)

Optional: fpmt (Rate, NPer, PV, FV)
fpmt (Rate, NPer, PV, FV, Due)

Parameters:

Rate is the interest rate per period. For example, if you get a car loan at an annual percentage rate (APR) of 6 percent and make monthly payments, the rate per period is $6/12$, or 0.5.

NPer specifies the total number of payment periods in the annuity. For example, if you make monthly payments on a four-year car loan, your loan has a total of 4×12 (or 48) payment periods

PV specifies the present value (or lump sum). For example, when you borrow money to buy a car, the loan amount is the present value

FV optional specifying future value or cash balance after final payment is made. For example, the future value of a loan is \$0 because that is its value after the final payment. However, if you want to save \$50,000 during 18 years for your child's education, then \$50,000 is the future value. If omitted, 0 is assumed.

Due optional specifies when payments are due. This argument must be either 0, if payments are due at the end of the payment period, or 1, if payments are due at the beginning of the period. If omitted, 0 assumed.

FPPmt specifying the principal payment for a given period of an annuity based on periodic fixed payments and a fixed interest rate.

Syntax: `fppmt (Rate, Per, NPer, PV,)`

Optional: `fppmt (Rate, Per, NPer, PV, FV)`
`fppmt (Rate, Per, NPer, PV, FV, Due)`

Parameters:

Rate is the interest rate per period. For example, if you get a car loan at an annual percentage rate (APR) of 6 percent and make monthly payments, the rate per period is $6/12$, or 0.5

Per specifies the payment period in the range 1 through ***NPer***.

NPer specifies the total number of payment periods in the annuity. For example, if you make monthly payments on a four-year car loan, your loan has a total of 4×12 (or 48) payment periods.

PV is the present value (or lump sum). For example, when you borrow money to buy a car, the loan amount is the present value.

FV optional specifying future value or cash balance you want after you have made the final payment. For example, the future value of a loan is \$0 because that is its value after the final payment. However, if you want to save \$50,000 during 18 years for your child's education, then \$50,000 is the future value. If omitted, 0 is assumed.

Due optional specifies when payments are due. This argument must be either 0, if payments are due at the end of the payment period, or 1, if payments are due at the beginning of the period. If omitted, 0 assumed.

FPV returns the present value of an annuity based on periodic, fixed payments to be paid in the future and a fixed interest rate.

Syntax: fpv (Rate, NPer, Pmt)

Optional: fpv (Rate, NPer, Pmt, FV)
 fpv (Rate, NPer, Pmt, FV, Due)

Parameters:

Rate is the interest rate per period. For example, if you get a car loan at an annual percentage rate (APR) of 6 percent and make monthly payments, the rate per period is $6/12$, or 0.5.

NPer specifies the total number of payment periods in the annuity. For example, if you make monthly payments on a four-year car loan, your loan has a total of 4×12 (or 48) payment periods.

Pmt specifying payment to be made each period. Payments usually contain principal and interest that does not change over the life of the annuity.

FV optional specifying future value or cash balance you want after you have made the final payment. For example, the future value of a loan is \$0 because that is its value after the final payment. However, if you want to save \$50,000 during 18 years for your child's education, then \$50,000 is the future value. If omitted, 0 is assumed.

Due optional specifies when payments are due. This argument must be either 0, if payments are due at the end of the payment period, or 1, if payments are due at the beginning of the period. If omitted, 0 assumed

FRate returns the interest rate per period for an annuity.

Syntax: frate (NPer, Pmt, PV)

Optional: frate (NPer, Pmt, PV, FV)
 frate (NPer, Pmt, PV, FV, Due)
 frate (NPer, Pmt, PV, FV, Due, Guess)

Parameters:

NPer specifies the total number of payment periods in the annuity. For example, if you make monthly payments on a four-year car loan, your loan has a total of 4×12 (or 48) payment periods.

Pmt specifying payment to be made each period. Payments usually contain principal and interest that does not change over the life of the annuity.

PV is the present value (or lump sum). For example, when you borrow money to buy a car, the loan amount is the present value.

FV optional specifying future value or cash balance you want after you have made the final payment. For example, the future value of a loan is \$0 because that is its value after the final payment. However, if you want to save \$50,000 during 18 years for your child's education, then \$50,000 is the future value. If omitted, 0 is assumed.

Due optional specifies when payments are due. This argument must be either 0, if payments are due at the end of the payment period, or 1, if payments are due at the beginning of the period. If omitted, 0 assumed.

Guess optional specifying value you estimate will be returned by IRR. If omitted, Guess is 10 percent

FSLN returns a value specifying the depreciation of an asset for a specific time period using the double-declining balance method or some other method you specify.

Syntax: `fsln (Cost, Salvage, Life, Period)`

Parameter:

Cost Specifying initial cost of the asset.

Salvage Specifying value of the asset at the end of its useful life.

Life Specifying length of useful life of the asset

FSYD returns the sum-of-years digits depreciation of an asset for a specified period.

Syntax: `fsyd (Cost, Salvage, Life, Period)`

Parameter:

Cost specifying initial cost of the asset.

Salvage specifying value of the asset at the end of its useful life.

Life specifying length of useful life of the asset.

Period specifying period for which asset depreciation is calculated.

Attachment

Keyboard short cuts

The keyboard inputs in this description correspond to the English keyboard and Windows regional and language option English-US. When using a non-English keyboard or language, some functions are acquired with other key combinations. This concern most of the **Ctrl** key functions. In the attachment of this manual you will find pictures about key codes of different keyboards. Read the description below about keyboard configurations.

You can type in letter of the alternative font by pressing the **Ctrl** key. Example: press **Ctrl+P** to write the character π or **Ctrl+L** to write the letter λ .

Enter	Exit escape mode
	Exit Superscript
	Exit Subscript
	Moves the cursor from end of the fraction bar to numerators first column.
	Moves the cursor from numerator to denominators first column.
	Moves the cursor from denominator to end of fraction bar.
Enter + Ctrl	Start calculation and display results.
Enter + Shift	Line feed- return : move the cursor to the first used column in the next row
Alt + Ctrl + O	Open an existing document
Alt + Ctrl + R	Reload the recent file
Alt + Ctrl + S	Save the active worksheet
Alt + Ctrl + Y	Redo - Repeats the last action
Alt + Ctrl + Z	Undo - Reverses the last action
Ctrl + ,	
Ctrl + _	Toggle on / off Subscript
Ctrl + Shift + ,	
Ctrl + 6	Toggle on / off Superscript (exponent)
Ctrl + 9	large round bracket open
Ctrl + 0	large round bracket close
Ctrl + [large square bracket open
Ctrl +]	large square bracket close
Ctrl + Shift + {	large curly bracket open
Ctrl + Shift + }	large curly bracket close
Ctrl + /	fraction line
Ctrl + 1	root
Ctrl + 2	Exponent 2
Ctrl + 3	Exponent 3
Ctrl + 4	Integral Formula
Ctrl + Shift + 4	Integral Symbol
Ctrl + 5	Function Symbol
Insert	Change from overwrite to insert mode
Insert + Ctrl	Inserts a space at the cursor position

Insert + Shift	Inserts a row at the cursor position
Delete	Deletes the character at cursor position
	If a range is selected, the selected range are deleted
Delete + Shift	Deletes a row at the cursor position
Home	Moves the cursor to the first used column
	Moves the cursor to the first column
End	Moves the cursor to the last used column
	Moves the cursor to the last column
Ctrl + Csr left	Reduce box size, deletes the column at the right edge
Ctrl + Csr right	Reduce box size, deletes the column at the left edge
Ctrl + Csr up	Reduce box size; deletes the row at the bottom edge
Ctrl+Csr down	Reduce box size, deletes the row at the top edge
F2	Mark the selected range as remark
F2 + Ctrl	Resets the selected range as remark
F3	Enables or disables the Superscript mode
F4	Enables or disables the Subscript mode
F6	Clears all
F7	Clears the output of the calculator
F8	Starts the calculator

		Shift		Additional functions
Ctrl + A	α	A	Alpha	
Ctrl + B	β	B	Beta	
Ctrl + C	χ	X	Chi	Copied the selected area *
Ctrl + D	δ	Δ	Delta	
Ctrl + E	ε	E	Epsilon	
Ctrl + F	ϕ	Φ	Phi	
Ctrl + G	γ	Γ	Gamma	
Ctrl + H	η	H	Eta	
Ctrl + I	ι	I	Iota	
Ctrl + J	φ		Phi (alt.)	
Ctrl + J		ϑ	Theta (alt.)	
Ctrl + K	κ	K	Kappa	
Ctrl + L	λ	Λ	Lambda	
Ctrl + M	μ	M	Mu	
Ctrl + N	ν	N	Nu	
Ctrl + O	\omicron	O	Omicron	
Ctrl + P	π	Π	Pi	
Ctrl + Q	θ	Θ	Theta	
Ctrl + R	ρ	P	Rho	
Ctrl + S	σ	Σ	Sigma	
Ctrl + T	τ	T	Tau	
Ctrl + U	υ	Y	Upsilon	
Ctrl + V	ϖ		Pi (alt.)	Insert text from clipboard **
Ctrl + V		ς	Sigma (alt.)	
Ctrl + W	ω	Ω	Omega	
Ctrl + X	ξ	Ξ	Xi	Cut and copies the selected area *
Ctrl + Y	ψ	Ψ	Psi	

Ctrl + Z ζ Z Zeta

Decimal key The decimal key on the numeric keypad produces a decimal point always, regardless of the country setting.

*) **Ctrl + C** copies the selected area to clipboard. **Ctrl + X** cuts the selected area and copies it to the clipboard. If no area is selected, the corresponding Greek letter is written.

) **Ctrl + V writes the text from the clipboard to the cursor position if, immediately before a text with **Ctrl + C / X** was copied, otherwise the corresponding Greek letter is written.

Text box short cuts

For editing of text the following table shows a list of keyboard instructions.

Keys	Operations
Ctrl + Tab	Tab
Ctrl + Number Pad 5	Select all
Ctrl + A	Select all
Ctrl + E	Center alignment
Ctrl + J	Justify alignment
Ctrl + R	Right alignment
Ctrl + L	Left alignment
Ctrl + C	Copy
Ctrl + V	Paste
Ctrl + X	Cut
Ctrl + Z	Undo
Ctrl + Y	Redo
Ctrl + '+'	Superscript
Ctrl + '='	Subscript
Ctrl + 1	Line spacing = 1 line.
Ctrl + 2	Line spacing = 2 lines.
Ctrl + 5	Line spacing = 1.5 lines.
Ctrl + ' (apostrophe)	Accent acute
Ctrl + ` (grave)	Accent grave
Ctrl + ~ (tilde)	Accent tilde
Ctrl + ; (semicolon)	Accent umlaut
Ctrl + Shift+6	Accent caret (circumflex)
Ctrl + , (comma)	Accent cedilla
Ctrl + Shift + ' (apostrophe)	Activate smart quotes
Backspace	Delete previous character.
Ctrl + Backspace	Delete previous word.
F16	Same as Backspace.
Ctrl + Insert	Copy
Shift + Insert	Paste
Insert	Overwrite
Ctrl + Left Arrow	Move cursor one word to the left.
Ctrl + Right Arrow	Move cursor one word to the right.
Ctrl + Left Shift	Left alignment
Ctrl + Right Shift	Right alignment
Ctrl + Up Arrow	Move to the line above.
Ctrl + Down Arrow	Move to the line below.
Ctrl + Home	Move to the beginning of the document.
Ctrl + End	Move to the end of the document.
Ctrl + Page Up	Move one page up.
Ctrl + Page Down	Move one page down.
Ctrl + Delete	Delete the next word or selected characters.
Shift + Delete	Cut the selected characters.
Ctrl + Shift + A	Set all caps.
Ctrl + Shift + L	Fiddle bullet style.

Ctrl + Shift + Right Arrow increase font size
 Ctrl + Shift + Left Arrow decrease font size

Key Code Configuration

US-English

~ `	1 !	2 @	3 #	4 \$	5 %	6 ^	7 &	8 *	9 (0)	- _	= +	Backspace
Tab	Q	W	E	R	T	Y	U	I	O	P	{ [}	 \
Caps Lock	A	S	D	F	G	H	J	K	L	:	"	Enter	
Shift	Z	X	C	V	B	N	M	<	>	? /	1 2	Shift	
Ctrl	Win Key	Alt							Alt	Win Key	Menu	Ctrl	

German

° ^	1 !	2 "	3 §	4 \$	5 %	6 &	7 /	8 {	9 (0)	=	? }	←
Tab	Q	W	E	R	T	Z	U	I	O	P	Ü	* +	↵
↓	A	S	D	F	G	H	J	K	L	Ö	Ä	' #	
↑	> <	Y	X	C	V	B	N	M	;	:	- _	Shift	
Strg	(Win)	Alt							Alt Gr	(Win)	(Menu)	Strg	

Italian

\	! 1 ✓	" 2 ✕	£ 3 ✕	\$ 4 ∫	% 5 f	& 6	/ 7 1/2	(8 () 9)	= 0	? ' i	^ X y	Backspace
Tab	Q	W	E € R	T	Y	U	I	O	P	é { }	* }	Enter	
			è []								+]		
Caps Lock	A	S	D	F	G	H	J	K	L	ç °	@ #	§ ù	
Shift	>	Z	X	C	V	B	N	M	;	:	- X y	Shift	
	<								,	.			
Ctrl	Win Key	Alt							Alt Gr	Win Key	Menu	Ctrl	

Brazil (Portuguese)

" '	! 1 ✓	@ 2 X	# 3 X	\$ 4 £	% 5 f	¨ 6 ¬	&	*8	(9 () 0)	- X y	+ 1/2	← Backspace
Tab ↔	Q / /	W ? ?	E € €	R	T	Y	U	I	O	P	· · ·	{ } []	Enter ↵
Caps Lock ⬆	A	S	D	F	G	H	J	K	L	Ç	^ ~ X y	} }]	
Shift ⬆	 \ \ /	Z	X	C ç	V	B	N	M	< ,	> .	: ;	? / /	Shift ⬆
Ctrl	Win Key	Alt								Alt Gr	Win Key	Menu	Ctrl