

RedCrab

The Calculator

Version 4.34 News

copyright © by Redchillicrab, Singapore 2009 - 2013

Version 4.34 news (User Manual)

5.9 Sqr

The *Sqr* function returns the square of the argument.

Example: `Sqr(4)` = 16

5.10 Sqrt

The result of *Sqrt* is the square root of the argument.

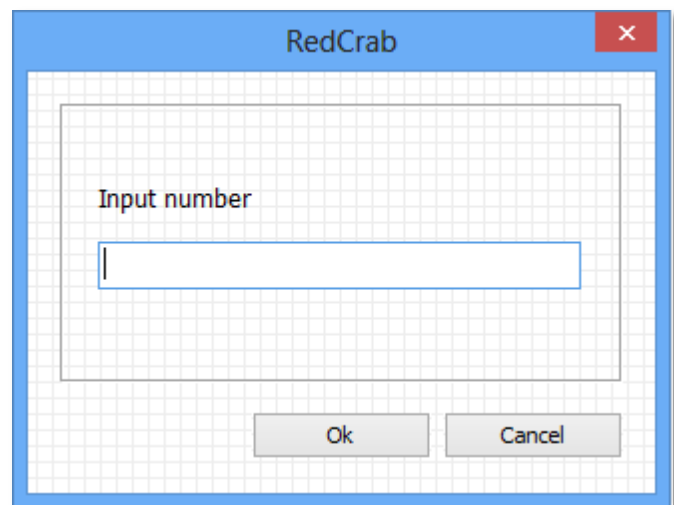
Example: `Sqrt(4)` = 2

Version 4.34 news (Programmers Manual)

3.1 Input

The *input* statement opens up a request window for input of a numeric value. The *input* statement follows two or three parameters:

1. A text string, which is shown above the input line.
2. The name of the variable, whose input value will assign to.



3. With a third string parameter, you can define an optional preset value, which is displayed in the input line when the window is shown.

Example 1: `input "Input number", x, "123"`

Example 2: `let a = "Input number"`
 `Let b = "123"`
 `Input a, x, b`

3.2 Display

Use the ***display*** statement to display a numeric value in a message window. The ***display*** statement follows two parameters:

1. The variable, whose value will be displayed in the window.
2. A text string, which can be displayed in addition to the value. The string can contains format instructions.

Example 1: `Let x = 471.2`
 `Display x, "The result is: "`

The message window show: „The result is: 471.2“

Example 2: `Let x = 471.2`
 `Display x, "The result is: #.##"`

The message window show: „The result is: 471.20“

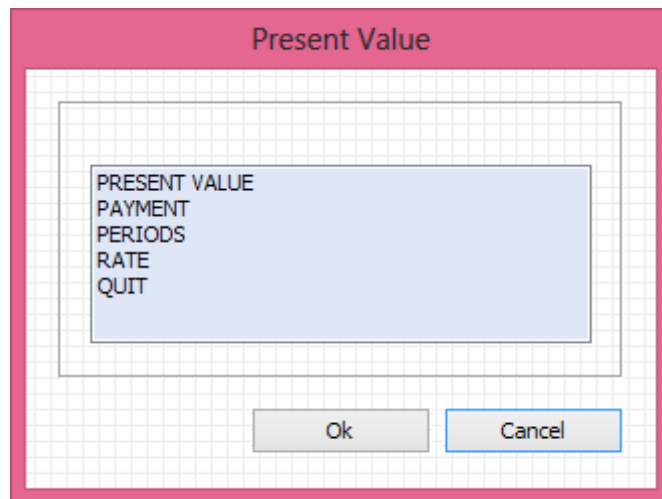
The format instructions are conform to the result box formatting. For more information, read the description in the user manual: ***Result Box / Formatting***.

3.3 Menu

With the *menu* statement you can open a menu window. The statement follows different numbers of text string parameters.

The first parameter is the menu title and follows by the name of the first menu item and the function, which is to call, if the user clicks this item. It follows the name of the second menu item and the calling function, and so on. All parameters are separated by comma and included in quotation marks.

Example: Menu "Present Value",
 "PRESENT VALUE", "PresentVal",
 "PAYMENT", "PresentPayment",
 "PERIODS", "PresentPeriods",
 "RATE", "PresentRate",
 "QUIT", "Quit"

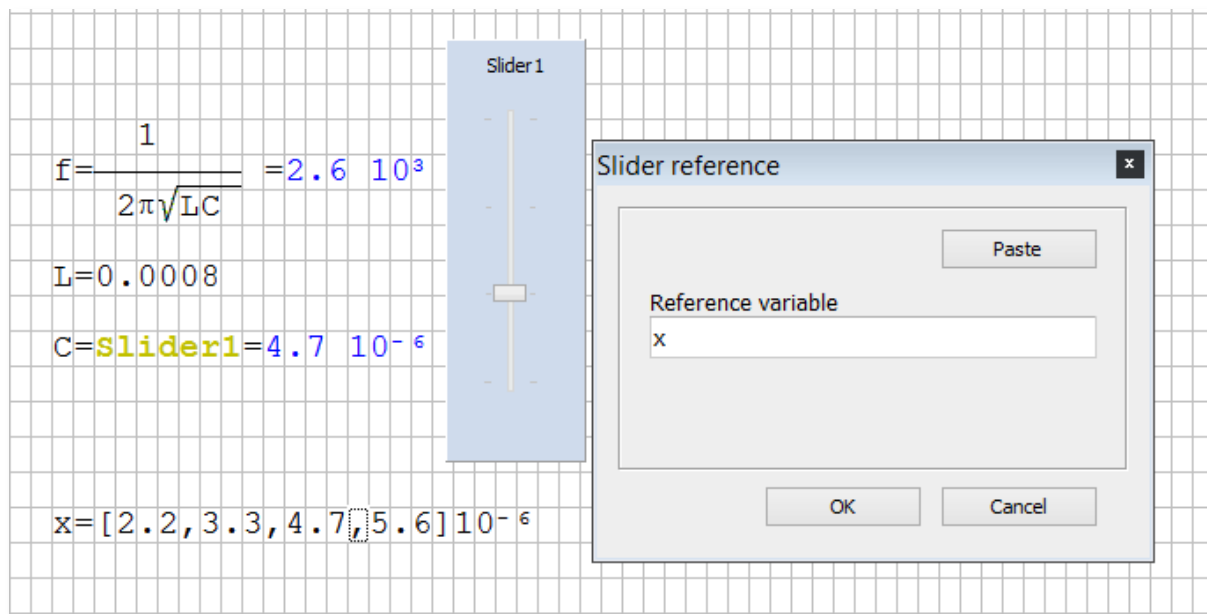


Version 4.32 news (User Manual)

3.6 Menu Insert Slider

The Menu **Insert.Slider** inserts a slider in the worksheet. You can use the slider like a variable in any expression. The adjustment of the slider changes the output value. The changed result of the expression will be refreshed automatically.

Example:



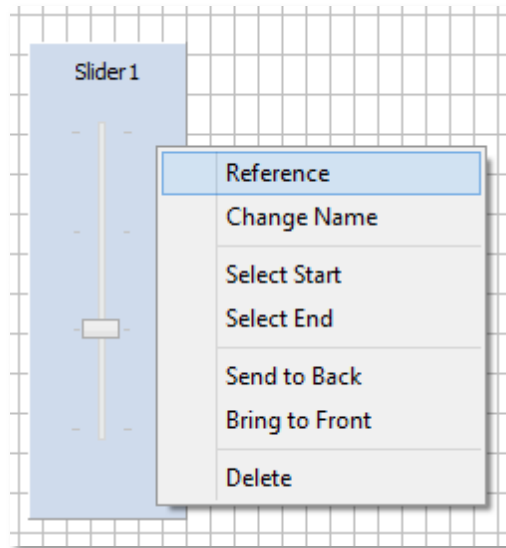
In the example above, the slider output value is assigned to the variable C . The range of the slider values is defined in the variable x . If the slider is adjusted, the numbers, on the right of Slider1 and the result of the expression will be refreshed automatically.

Slider Popup Menu

Reference opens a dialog box to input the name of the reference variable. The reference variable contains a data field with the values, which can be selected with the slider.

The name can be inserted with **Paste** button or entered per keyboard. The keyboard provides here **ANSI** characters only. Variable names which included Greek characters must be inserted with **Paste** button.

Instead the name of a variable, it is possible to set a reference to an imported text file, which contains a data field.



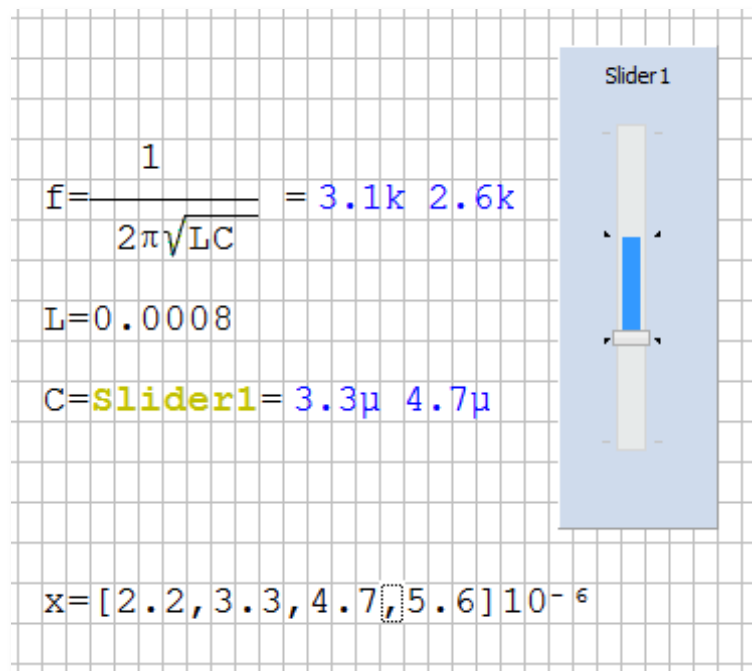
Change Name

Change Name opens a dialog box to change the name of the slider.

Select Range

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 **Select Start**. Then move the slider button to the second position and click on **Select End**.



The image above shows an example with the selected values 3.3 and 4.7. On the right of the expression, **RedCrab** displays the results for both input values.

Version 4.32 news (Programmer Manual)

2.3 Define

(..... Extension)

The expression can be a value, a variable, a data field, a function that returns a value or several of these. The example below defined *x* as a data field with 20 rows and 8 columns.

Example: `define x[] = [1..20] * [1..8] fill 0`

2.4 Let

(..... Extension)

Example:

```
let x[5] = 16
```

The example above shows the last row assigned the value of 16 to index [5] of *x*. Index [5] is the sixth element of the field. The first element is index [0].

2.13Next

Use *Next* to assign a data series to a field variable. You can assign single value or single row to the field row by row. The special feature is that *Next* needs no index. The field variables managed the index handling. Every execution of *Next* increments the index by one.

Beispiel: `define x = [1..20]`

```
next x
next x = 23
next x = 5.6
```

In the example above, *Next* x initialize the index of x . The next rows assign 23 to $x[0]$ and 5.6 to $x[1]$. If you pass x as an argument to a function or another variable, the index will be pass on.

Another feature of *Next* is the range control. If the index reaches the tail of the data field, *Next* extend the data field automatically. However, for big data fields, it is useful to define the data field large enough. The late extension of the field needs more processing power. For small data fields up to few thousand records, this is irrelevant. In any case, it is important that you define a variable with a minimum of one record and the number of columns you need.

Next checks the data compatibility. The dimension of the assigned data field must be one less the field variable, or the same as one row of the field. The example above assigned a simple value to a one-dimensional data field. In the example below, we move a one-dimensional field to a row of a two-dimensional field.

The numbers of columns can be different. The example below defined x as a two-dimensional field that contains 3 columns. In row 4 we assign a one-column field with the value of 99 to x . The remaining columns are filled with 0. In row 5 we assign a four-column field to x . *Next* cuts the fourth element.

Example: `define x[] = [1..6]*[1..3] fill 1`
`next x`
`next x = [10, 20, 30]`
`next x = [99]`
`next x = [22, 33, 44, 55]`

Content of x :

10	20	30
99	0	0
22	33	44
1	1	1
1	1	1
1	1	1

2.14 Index

The function ***Index*** returns the current index of a field variable (the index of the last ***Next*** move).

Example: `i = index(x)`

Version 4.31 news (Free + Shareware)

The reference variable of result boxes can now be input with keyboard.

The reference variable of result boxes can now be inserted with keyboard instead the *Paste* button. Please note that the keyboard provides *ANSI* symbols only. Names which include Greek characters must be inserted per *Paste* button.

Undo function improved

Undo deletes function names, when inserted by button click, completely (not character by character).

Bug fixes

Chart box becomes invisible, when toggled between *NonSync* and *GridSync* mode.

After loading of *RedCrab*^{PLUS} demo files in freeware mode, there was a problem with saving of freeware files.

The definition of complex data field which includes data ranges could cause a runtime error.

New features in Version 4.31 (RedCrab^{PLUS} only)

3.4 Menu View Program Panel

When the option Program Panel in menu View is switched on, **RedCrab** creates a button for any function in the program editor. A single click on the button inserts the function name in the worksheet on the actual cursor position. A double click inserts the name and the parameter list.

If the program changed, you can refresh the button list with Refresh in the popup menu. To open the popup menu, click with the right mouse button in the button area.

If the button panel too large for the window, you can move it with the left mouse button.

