

UCALC Overview

UCALC contains various tools for performing calculations, all combined into one powerful yet simple program. Here's a highlight of some of the features:

- **Expression Evaluator** which supports a number of built-in functions, operators, numeric bases, numeric formats, and modes.
- **General Ledger**. UCALC is not just for heavy duty scientific calculations. This simple general ledger provides a convenient way for adding up many numbers.
- **Unit Converter**. Allows you to convert between units of measure in a flexible manner. It also includes a world currency unit category.
- **User Solution Modules**. This innovative feature allows users to interactively define and solve problems by simply filling in the blanks without having to manipulate math formulas.
- **User Functions & Variables**
- **Equation Solver** which allows you to solve for the unknown value in an algebraic expression.
- **Summation Table** is a general mathematical tool which can be used for practical things such as calculating annuity, compound growth, etc...
- **Numerical Integration**. A calculus tool.
- **Graphing**. UCALC can plot Cartesian, polar, parametric, 3D equations, or coordinates from data files.
- **

** This last space is intentionally left blank. It represents features that you would like to see in a future version of UCALC. Much of the way UCALC currently operates is a direct result of suggestions from a number of users like you. If there is a feature that you would like to see next time, let the [author](#) know about it.

Disclaimer

This software is provided on an "as is" basis without warranty of any kind, expressed or implied. The person using this software bears all risk as to its quality and performance. The author will not be held liable for any special, incidental, consequential, direct or indirect damages due to any malfunctions.

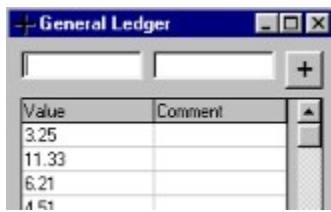
Help file generated by VB HelpWriter.


General Ledger

See also [Summation Table](#)

To activate the *general ledger*, either click on  or select **General Ledger** from the **Tools** menu.

UCALC includes a simple *general ledger* for adding a list of numbers. Let's say you have a long phone bill, and you wanted to add up all the calls you placed to New York to see if that's the most expensive part of the bill. You might not be able to fit all of these numbers on the same line using the *expression evaluator*. Instead, it will probably be more convenient to do it in the column format provided by the *general ledger*.



Simply type in a numeric value (or expression) in the upper left blank, then either press **[enter]** or click on  to enter the next number. UCALC recalculates the total as you go along. Next to each value, you have the option of entering a comment. To delete an entry, simply highlight it, and press the **[Del]** key. To start over with an empty table, click on the **clear** button at the bottom.




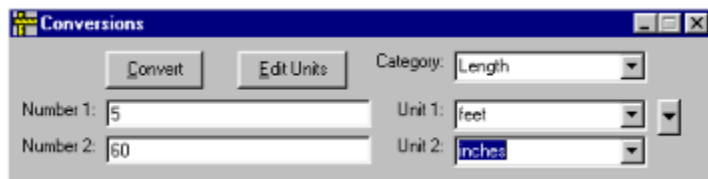
To subtract a number, simply make it a negative number. For instance you might enter -10.53 to subtract 10.53.

You can [save](#) your calculations to a file and load them later.

Help file generated by VB HelpWriter.


Unit Conversions

To activate the unit converter, either click on  from the toolbar, or select **Unit Converter** from the **Tools** menu. The flexible converter allows various ways of performing conversions. The general procedure works like this:



1. Select a category by clicking on the **Category** combo box.
2. Enter a number where it says **Number 1**.
3. Select the unit to convert *from* where it says **Unit 1**.
4. Select the unit to convert *to* where it says **Unit 2**.
5. The result is displayed where it says **Number 2**.

This is simply one way to do it. It can also be done by entering an initial value where it says **Number 2** instead. Or you may enter the units first, and a value in any *Number* field. Then click on the **Convert** button.

To add more fields, click on  (on the far right). This allows you to convert from one unit to as many others as needed.

Defining your own units:

UCALC comes with a large number of units under a variety of categories such as Length, Temperature, Time, etc..., and even monetary currencies from around the world. You can easily add to the list, or edit what is currently there. To do it, either click on the **Edit Units** button, or select **Conversions** from the **Define** menu.

First, select a category from the **Unit Category** combo box. If you would like to create a new category, select **{Create a new category}** and enter a name for it. Then simply fill in the blanks. Where it says **Unit 1** and **Unit 2**, enter unit names. Where it says **A ==> B**, and **B ==> A**, enter the formulas for the conversions. Press **Ok** to accept a new unit before moving on. To continue adding new units, fill in the empty blanks again as before. To edit an existing conversion, simply click on it, and modify the fields at the top.

Unlike other unit conversion programs which only allow you to define coefficient constants for user defined units, with UCALC you can define the new units with actual formulas. See the *Temperature* category as an example. When defining a new unit, simply enter the formula which relates it to any previous unit. The formula must be defined using the variable **x**. UCALC will then automatically know how to convert this new unit to all the other units already defined.

Note: The order of the placement of units on the list does not matter, however there should be no duplicate or circular definitions. Also each unit must be defined so that it relates either directly, or indirectly to every other unit in the same category.

Error Example 1:

feet ---- inches
inches ---- miles
miles ---- meters
meters ---- inches

Here the last definition (meters - inches) is unnecessary. The first three lines are sufficient for converting between all of the units involved in that list.

Error Example 2:


feet ---- inches
inches ---- miles
meters ---- kilometers

In this example, there is a missing link. A definition must be added which relates either *meters* or *kilometers* to one of the previous units.

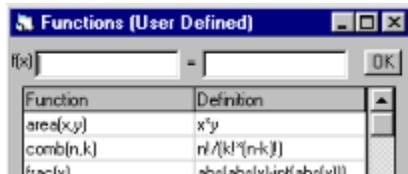
Help file generated by VB HelpWriter.

User Defined Functions

See also [Built-in Functions](#)
and [User Defined Variables](#)

To define user functions, either click on  from the toolbar, or select **Functions** from the **Define** menu. Functions can also be defined from the *expression evaluator*.

A function definition consists of a *function name*, one or more *arguments* (optional), and an *expression* which represents the definition of the function.



Enter a new function name in the upper left text box. For instance *myfunction(a,b,c)*. Then enter the definition in the text box on the right, based on the arguments, for instance $a*b+c$. Then click on **Ok**. Repeat the procedure to define more functions. To modify an existing function, click on it, modify it, and then click on **Ok**. To delete an existing function, click on it then press the **[Del]** key.

To add a text explanation of the function, after the definition enter the character `;` (semicolon) followed by some text. In order to see comments, maximize the function definition window.

To make UCALC recognize these new definitions only during the current UCALC session, click on the **Apply** button. Once UCALC is unloaded, these definitions will be forgotten (unless *auto session recall* is on). To make the changes permanent, click on **Save**. (You can always modify or erase definitions that you have saved). When saving functions, variables are also saved.

You may also enter the function definition directly in the *expression evaluator* as well. For instance, you can enter: $myfunction(a,b,c) = a*b+c$.

Functions and arguments can have any name. The first character at least must be a letter of the alphabet. UCALC is not case sensitive. So *MyFunction* is the equivalent of *MYFUNCTION* or *myfunction*. A function may include other functions and variables in the definition as long as they are already defined. A function's arguments are prioritized over variables of the same name. So $myfunction(a,b)=a+b$, is not affected by variables of the same name which may have been defined.

Examples:

$Area(length,width) = length*width$

$frac(x)=abs(abs(x)-int(abs(x)))$; Fractional part of a number

$myfunction() = last*2$; Arguments are optional.

Help file generated by VB HelpWriter.

Built-in Functions and Operators

See also [User Defined Functions](#)
and [Order of Precedence](#)

Symbol	Equivalent	Description	Example
()		Prioritizes an expression	$5*(1+1) = 10$
!	FACT	Factorial	$5! = 120$
%		Percentage	$35\% = 0.35$
^	**	Raised to the power of	$4^5 = 1024$
*		Multiply by	$3 * 6 = 18$
/		Divide by	$9 / 2 = 4.5$
MOD		Modulo (remainder)	$7 \text{ MOD } 4 = 3$
+		Add	$1 + 1 = 2$
-		Subtract	$9 - 5 = 4$
>		Greater than	$9 > 2 = 1$
<		Less than	$7 < 4 = 0$
==	=	Equal test	$5 == 4 = 0$
>=	>=	Greater or equal	$3 >= 3 = 1$
<=	<=	Less or equal	$\#h3E <= 9 = 0$
<>		Not Equal	$\#b10101 <> 20 = 1$
NOT		Bitwise 'not'	$\text{not}(15) = -16$
AND	&	Bitwise 'and'	$\#b101 \text{ and } \#h1E = 4$
OR		Bitwise 'or'	$13 \text{ or } 6 = 15$
XOR		Bitwise 'exclusive or'	$9 \text{ xor } 3 = 10$
EQV		Bitwise 'equivalence'	$6 \text{ eqv } 9 = -16$
IMP		Bitwise 'implication'	$1 \text{ imp } 5 = -1$
SIN		Sine	$\sin(\pi) = 0$
COS		Cosine	$\cos(\pi) = -1$
TAN		Tangent	$\tan(\pi) = 0$
ASIN		Arcsine	$\text{asin}(1) = 1.570$
ACOS		Arccosine	$\text{acos}(-1) = 3.141$
ATAN	ATN	Arctangent	$\text{atan}(0) = 0$
SINH		Hyperbolic sine	$\sinh(3) = 10.01$
COSH		Hyperbolic cosine	$\cosh(2) = 3.76$
TANH		Hyperbolic tangent	$\tanh(1) = 0.76$
COTH		Hyperbolic cotangent	$\text{coth}(1) = 1.31$
SECH		Hyperbolic secant	$\text{sech}(0) = 1$
CSCH		Hyperbolic cosecant	$\text{csch}(1) = 0.85$
ASINH		Hyperbolic arcsine	$\text{asinh}(2) = 1.44$
ACOSH		Hyperbolic arccosine	$\text{acosh}(9) = 2.89$
ATANH		Hyperbolic arctangent	$\text{atanh}(.1) = 0.10$
ACOTH		Hyperbolic arccotangent	$\text{acoth}(7) = 0.14$
ASECH		Hyperbolic arcsecant	$\text{asech}(.3) = 1.87$
ACSCH		Hyperbolic arccosecant	$\text{acsch}(2) = 0.48$
ABS		Absolute value	$\text{abs}(-8) = 8$
EXP		e to the power of	$\text{exp}(3) = 20.08$

RND		Random number	rnd(1) = .969
INT		Truncate to an integer	int(6.8) = 6
CEIL		Round upward	ceil(6.8)= 7
EXP2		2 to the xth power	exp2(3) = 8
EXP10		10 to the xth power	exp10(3) = 1000
FACT	!	Factorial	fact(5) = 120
LOG2		Log base 2	log2(8) = 3
LOG10		Log base 10	log10(100) = 2
LOG	LN	Natural log	log(16) = 2.77
SGN	SIGN	Sign of expression (-1,0, or 1)	sgn(-9) = -1
SQR	SQRT	Square root	sqr(64) = 8

Note: Function parameters must always be enclosed within parenthesis. Also, the opening parenthesis is part of the function name. There should be no space or operator between the beginning of the function name and the opening parenthesis.

Examples:

Incorrect	Correct	Comment
sin 5x	sin(5x)	Parentheses are required
sin (3x)	sin(3x)	There should be no space between sin and (
sin^2(x)	sin(x)^2	Nothing should be placed between sin and (

Help file generated by VB HelpWriter.

Expression Evaluator

To activate the *expression evaluator*, select **Expression Evaluator** from the **Tools** menu.

UCALC's *expression evaluator* can serve a purpose similar to that of a conventional hand held calculator. However, it uses a very different approach which is more natural to the way people actually calculate. For instance, if you wanted to calculate $10+3 - 2$ on a conventional calculator, each time you entered a new number you would lose sight of the previous one. If you made a mistake in the middle of a long operation, you might have to start all over. Some calculators attempt to simplify the process by providing memories, stacks, registers, and other such devices, all of which can be complicated.

With the *expression evaluator* provided by UCALC, mathematical expressions such as $10+3 - 2$ are entered just the way you would write them on paper. This allows you to see all of what you are calculating. With this, there is no fear of entering long and intricate expressions, because they can be edited.

Instead of using complicated memories and registers, UCALC allows you to define variables with any name you choose. UCALC also has many built-in functions and operators which allow you to perform intricate scientific calculations. You can also easily create your own user-functions.

The *expression evaluator* supports numeric formats which can either be chosen from a list, or custom designed. It also has several angle modes useful for trigonometric operations.

The *expression evaluator* returns numeric answers to expressions entered in algebraic form.

Examples:

$5+4 - 3$

$20\% * 137.25$; 20% here is treated as 0.20

$\#b10011001$ and $\#b10100011$; See Numeric Bases

$(\text{abs}(\sin((3+2^{\log(1+8)})))/(5+5.4)))^{(2\pi)+1/((\pi+1)(\pi-1))} * 17$

$\text{price} - \text{discount}$; assuming you have defined *price* & *discount* as variables

$3 + \text{area}(5,2) * \text{area}(\sin(\pi)+5, \text{area}(a,b))$
; assuming you have defined *area* as a user function, and *a* & *b* as variables

For the following examples, you can either enter them in the *expression evaluator*, or use the indicated specialized feature as an alternative:

$\text{sum } x^2+5, 1..10$; To view a grid of values, use Summation Table


$\text{MyVariable} = 100$; See also User Variables

$\text{myfunction}(a,b,c,d) = a+b*c-d$; See User Functions

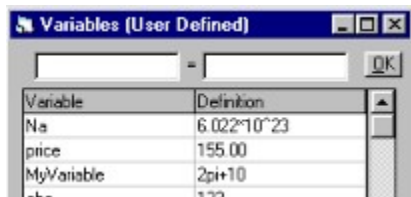
Help file generated by VB HelpWriter.

User Defined Variables

See also [Built-in Functions](#)
and [User Defined Functions](#)

To define user variables, either click on  from the toolbar, or select **Variables** from the **Define** menu. Variables can also be defined from the *expression evaluator*.

A variable definition consists of a *variable name*, and an *expression* which evaluates to a number that this variable will hold.



Enter a new variable name in the upper left text box. For instance *myvariable*. Then enter a number, or an expression which evaluates to a number, in the text box on the right. Then click on **Ok**. Repeat the procedure to define more variables. To modify an existing variable, click on it, modify it, and then click on **Ok**. To delete an existing variable, click on it then press the **[Del]** key.

To add a text explanation of the variable, after the numeric value enter the character `;` (semicolon) followed by some text. In order to see comments, maximize the variable definition window.

To make UCALC recognize these new variables only during the current UCALC session, click on the **Apply** button. Once UCALC is unloaded, these definitions will be forgotten (unless *auto session recall* is on). To make the changes permanent, click on **Save**. (You can always modify or erase definitions that you have saved). When saving variables, functions are also saved.

You may also enter the variable definition directly in the *expression evaluator* as well. For instance, you can enter: *myvariable = 100*.

Variables can have any name. The first character at least must be a letter of the alphabet. UCALC is not case sensitive. So *MyVariable* is the equivalent of *MYVARIABLE* or *myvariable*. A variable may include other variables in the definition as long as they are already defined. Although it is not recommended, a variable can share the same name as an unrelated function.

Examples:


Na = 6.022*10^23 ; This is Avogadro's number

MyVariable = 100

abc = 123 - MyVariable

Note: Constants **pi** and **e** are built-in, and their values cannot be changed. The independent variable (**x** by default) and the designated last variable (**last** by default) behave differently than user defined variables. Their values generally change from operation to operation.

User Solution Modules

For User Solution Modules, click on  from the toolbar, or select **User Solution Modules** from the **Tools** menu.

Once created, USMs allow you to interactively solve problems by simply filling in the blanks on a form, without having to directly manipulate math formulas. Several useful financial USMs are included to get you started. USMs are easy to create or modify.

When you first click on **USM**, you will be presented with the following box:



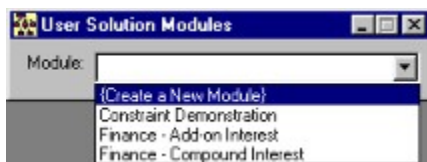
Click on the  symbol at the right of the **Module** combo box to select a pre-existing USM.



After that, part of the screen may look something like the example above. Simply click the option next to the field you want to find. Then fill in the other blanks with numeric values (or expressions). After that, click on the **Solve** button to find the answer. Any field can be selected as the unknown.

Note: When entering numbers, do not use currency symbols and *thousand separators* if the number is part of an expression. For instance \$2,000 by itself is acceptable in a currency field (in U.S. locale), but not \$2,000+\$5.00. Instead enter 2000+5.

Creating a New Module:



If you would like to create a new module, select **{Create a new module}** from the list. You will be prompted to enter a name for the Module. After entering a new name click on **OK** (or press **Enter**). Then you must enter a formula in the box that is labeled *Formula*. For instance:



When you click on **Continue**, UCALC will automatically interpret the formula and create the proper number of fields as follows:

Caption	Numeric Format	Constraints
LENGTH	General	
WIDTH	General	
HEIGHT	General	
VOLUME	General	

At this stage, by simply clicking on **Save** you will already have a USM that is ready to be used. Before saving, you have the option of modifying the properties in each field. (If you save it, you can always come back and modify it later by clicking on **Edit**).

UCALC originally places the name of a variable parameter as the default caption for each field. You may replace each caption with any description you prefer. Each field can also have its own Numeric Format, and constraints. Examine the financial USMs for examples of this.

Note: Keep in mind that when placed in the formula, built-in constants **e** and **pi**, as well as previously defined user variables are treated as numeric values instead of variable parameters. The independent variable (**x** by default), and the designated *last* variable (**last** by default) cannot be used in the formula as variable parameters.

Constraints:


Constraints are added to fields to prevent them from either accepting certain values, or solving for these values. For instance, you may not want the Volume of the rectangle to return or accept negative numbers, or you may want to accept only *integer* values in another field, or values between 10 and 20. Here are examples of valid constraints:

Constraints	Results that are accepted
x>0	Values greater than 0.
x<=0	Values smaller or equal to 0
x=int(x)	Only integer values
5..20	Only numbers between 5 and 20
x<>30	Any number except 30
10..50,x<>15	Numbers between 10 and 50 except 15
x<3 or x >9	Numbers smaller than 3 or greater than 9 but nothing in between

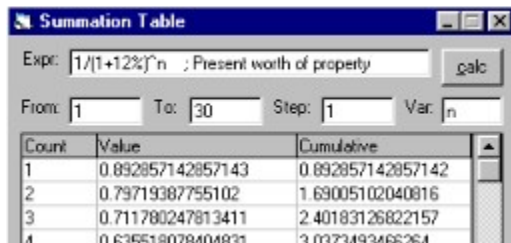
Note: The constraint field can accept any valid UCALC expression, and can be combined in many ways. The only thing to remember is that constraints must be defined based on the variable **x**. Select and run the *Constraint Demonstration* USM to see how constraints work, then edit it to examine how it was defined.

Summation Table

See also [General Ledger](#)

To use the *summation table*, either click on  from the toolbar, or select **Summation Table** from the **Tools** menu.

The *summation table* is a general mathematical tool which can be applied to practical calculations such as tabulating the present worth of an annuity, as in the example below:



Count	Value	Cumulative
1	0.892857142857143	0.892857142857142
2	0.79719387755102	1.69005102040816
3	0.711780247813411	2.40183126822157

This example uses the summation table to tabulate the present worth of a future amount of \$1 at 12% interest per year. Here, the *Count* column represents the number of years, the *Value* column represents the present worth of \$1, and the *Cumulative* column represents the present worth per year. The *Expr* field contains a formula for present worth. The line below indicates that it is going *From 1 To 30* years in increments (*Step*) of 1, using *Variable* n.

To simply return the total cumulative value of an expression, without using a table, use the *sum* command in the *expression evaluator*. The syntax is as follows:

```
sum expression ,a..b [,var [,step]]
```

Examples:


```
sum x^2,1..10
```

```
sum n+5,1..10,n,0.5
```

Help file generated by VB HelpWriter.

Numerical Integrations

See also [Expression Evaluator](#)

To do *numerical integrations*, either click on  from the toolbar, or select **Numerical Integrations** from the **Tools** menu. Integrations can also be entered in the *expression evaluator* following the *integ* command.

UCALC approximates the value for the definite integral of a function using Simpson's rule. It works properly only on continuous intervals. When using the toolbar option, enter the mathematical expression, then fill in the *From* and *To* fields below. *Var* represents the variable name you are using in the expression. *Precision* represents the number of subdivisions, which must be an even number. (If you enter an odd number, UCALC adds 1 internally to make it an even number in order to do the calculation).

To manually perform integrations without using the toolbar option, use the following syntax:

```
integ expression ,a..b [,var [,precision]]
```

Where **a..b** is the interval, and **var** is the variable name. The default variable name is **x**.


Example:

```
integ x^2, 2..5
```

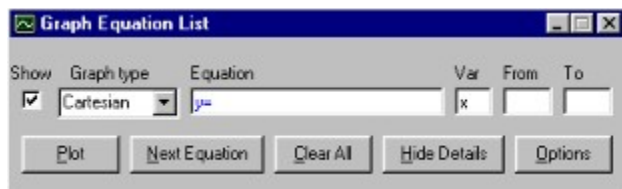
Note: When using the *numerical integration* option from the toolbar, enter only the expression to integrate, without the word *integ* or the extra parameters.

Help file generated by VB HelpWriter.

Graphing

To plot graphs, either click on  or select **Plot** from the **Graphing** menu.

UCALC can plot Cartesian, Polar, Parametric, 3D graphs, or X-Y coordinates from a data file. Up to 16 equations can be plotted together on the screen.



To plot equations, simply fill in the spaces above.

Show

The check box on the left determines whether or not that equation is plotted with the others.

Graph type

This combo box lets you choose which type of equation to plot. The categories are *Cartesian*, *Polar*, *Parametric*, *3D Surface*, and *Data File*.

Equation

A starting prompt is displayed depending on the *graph type* that is chosen; for instance **y=** for Cartesian. Enter your equation following the prompt.

Var

This is the name of the variable used in your equation.

From, To

This represents the starting and ending points for each equation. If left blank, Cartesian graphs will go from the default minimum, to the default maximum, or from the left to the right of the visible screen.

To add other equations, click on **Next Equation**. To add a graph title, labels, and to change some of the settings, click on **Options**. When ready to plot, click on **Plot**.


Help file generated by VB HelpWriter.

Numeric Formats

(Pre-defined)

See also [User Defined Formats](#),
[Numeric Bases](#)



To select or create a numeric format, go to , which is located towards the bottom right (if *Show Numeric Options* is not turned off). You can either select one of the nine pre-defined formats or create your own. This affects only the appearance of the output.

General

Displays the number as is. If no format is selected, then this is the one that goes into effect.

Example:

Input: 12345.12345
Output: 12345.12345

Currency

Displays numbers in the form of currency. The actual currency format depends on your computer's locale setting.

Example:

Input: 12345.12345
Output: \$12,345.12 (US locale)

Fixed

Displays one or more digits to the left of the decimal separator, and exactly 2 digits to the right.

Example:

Input: 12345.12345
Output: 12345.12

Standard

Uses a *thousand separator* when appropriate, and one or more digits to the left of the decimal separator, and exactly 2 digits to the right.

Example:

Input: 12345.12345
Output: 12,345.12

Percent

Displays the value of a number multiplied by 100, followed by the percent symbol (%), with exactly 2 digits following the decimal separator.

Example:

Input: 0.75
Output: 75.00%

Scientific

Standard scientific notation. Exponential notation with exactly 2 digits following the decimal separator.

Example:

Input: 12345.12345
Output: 1.23E+04

Yes/No

Displays *No* for 0, and *Yes* for all other values.

True/False

Displays *False* for 0, and *True* for all other values.

On/Off


Displays *Off* for 0, and *On* for all other values.

Note: Currency symbols and thousands separators are used only in the output. Generally, do not use them when entering numbers. They are accepted in User Solution Modules as long as the entered number is not part of an expression.

Help file generated by VB HelpWriter.

Equation Solver

See also [Expression Evaluator](#)

To solve an equation, either click on  from the toolbar, or select **Equation Solver** from the **Tools** menu. Equations can also be entered in the *expression evaluator* following the *solve* command.

The solver uses a recently enhanced adaptation of the Bisection Method algorithm, which makes it capable of finding one or more solutions to many types of equations. If you click on the toolbar option for solving, you are presented with options which can be changed.

The *equation solver* allows you to solve for the unknown in equations such as the following.

Examples:

$x^2=25$
 $2^x=x^2$
 $(x-5)(x-3)=100$
 $3x^2+x-5 = 27+x$
 $2^{(x+5)}=2^{17}$
 $(x+3)^2$; equivalent of $(x+3)^2=0$

For equations with multiple answers, you can narrow down the search. For instance, if you are solving $x^2=25$, and you are only interested in the negative answer, then change the second option to 0. It will then find a solution between -100000000 (if that is your default) and 0.

You can also solve directly by entering the equation in the *expression evaluator* after typing the word *solve*. For instance at the regular prompt, you may enter.

```
solve x^2=25
```

The full syntax for this kind of entry is:

```
solve equation [,a..b [,var]]
```

Where **a..b** is the interval in which you are searching for a solution, and **var** is the variable name for the unknown in your equation. Parameters following the equation are optional. The default name for the unknown is **x**. But it can be changed. For instance:

```
solve n^2=25,0..10,n
```

Note: When using the *equation solver* option from the toolbar, enter only the equation, without the word *solve*, or the extra parameters.

Help file generated by VB HelpWriter.

Order Of Precedence

See also [Built-in Functions](#)

Here is the precedence list from highest to lowest priority:

Anything inside parenthesis is performed first	()
Factorial, percentage	!, %
Exponentiation	^
Multiplication, division	*, /
Modulo (remainder)	MOD
Addition, subtraction	+, -
Relational operators	<, >, >=, <=, =, <>
AND operator	
OR, XOR (exclusive or)	
EQV (equivalence)	
IMP (implication)	

When consecutive operators have the same priority, UCALC evaluates from left to right. This means that an expression such as "a-b-c" is evaluated as "(a-b)-c".

Saving/Opening/Printing

See also [Special Copy](#)

UCALC allows you to save or print what you are doing in many different ways. If you would always like to start UCALC up where you left off the last time you exited the program, click on **Auto session recall** in the **Options** settings.

UCALC will save, print, or load your work according to what you are currently doing. For instance, if you go to the **Files/Save** menu while you are plotting graphs, you will have the option of saving these graphs either as a graphic file, or as a data file of coordinates. If you are entering equations for the graph, you will have the option of saving the list of equations. If you are working on the *general ledger*, you will be able to save the table and load it at another time. Each item saves, loads, or prints in its own way.

Help file generated by VB HelpWriter.

Numeric Formats

(User Defined)

See also [Pre-defined Formats](#)
[Numeric Bases](#)

To select or create a numeric format, go to the **Numeric Format** combo box at the bottom right of the program (if *Show Numeric Options* is not turned off). You can either select one of the nine pre-defined formats or create your own. This affects only the appearance of the output.

User formats consist of a combination of place holders, separators, formats, characters, and value dependent sections.

Place Holders

When the character **0** is used as a place holder, it displays a digit at that location if there is one, or 0 if there is none. This produces leading or trailing 0's when there are fewer digits than there are 0's to the right or left of the decimal separator. If there are more digits to the right than there are 0's, the decimal part is rounded to the number of 0's. When there are more digits on the left, they are unmodified.

Examples:

<i>Format</i>	<i>Input</i>	<i>Output</i>
00.00	1.5	01.50
00.00	1234.1234	1234.12

When **#** is used as a place holder, it displays a digit at that location if there is one, or nothing (instead of 0) if there is none. This produces numbers with *no* leading or trailing 0's. #'s to the right of the decimal separator correspond with the maximum number of digits on the right.

Examples:

<i>Format</i>	<i>Input</i>	<i>Output</i>
##.##	0.5	.5
##.##	1.5	1.5
##.##	1234.1234	1234.12

The **decimal separator** used in these examples is a point (.), based on the US locale. In some other locales a comma (,) is used instead.

The % is used as a **percentage** place holder. It multiplies the output by 100. When used in the input, it divides the number by 100.

Examples:

<i>Format</i>	<i>Input</i>	<i>Output</i>
##%	0.25	25%
##%	25%	25%

The **thousand separator** -- which is a comma (,) in the U.S. locale, or a point (.) in some other locales --

can be used between digit place holders to produce outputs such as 1,500,000. It can also be used just before the decimal separator or at the end, to scale down very large numbers.

Examples:

<i>Format</i>	<i>Input</i>	<i>Output</i>
#, #	1250000	1,250,000
#, #,	1250000	1,250
#, #,,	1250000	1

Scientific Format

To display numbers in *scientific* format, place **E-** (or **e-** for lowercase output) between place holders. **E+** (or **e+**) inserts a + sign for positive exponents.

Examples:

<i>Format</i>	<i>Input</i>	<i>Output</i>
00.00E+00	123456	12.35E+04
00E+00	123456	12E+04
0.00e-0	123456	1.23e5

Exponential notation can be used in the input as well.

Text

To display text as part of the output, enclose it within quotes.

Examples:

<i>Format</i>	<i>Input</i>	<i>Output</i>
#,.,# "million"	1234567	1.2 million
"\$"#,#.00	1234567	\$1,234,567.00

Value Dependent Sections

UCALC can use different formats depending on whether the answer is *positive*, *negative*, or *zero*. This is done by creating format sections separated by a semicolon (;). With two sections, the first one applies to positive numbers and 0, and the second one to negative numbers. With three sections, the first applies to positive, second applies to negative, and third applies to 0.

Examples:

<i>Format</i>	<i>Input</i>	<i>Output</i>
#;"("#)";"Zero"	123	123
#;"("#)";"Zero"	-123	(123)
#;"("#)";"Zero"	0	Zero

Note: Currency symbols and the *thousand separators* are used only in the output. Generally, do not use them when entering numbers. They are accepted in User Solution Modules as long as the entered number is not part of an expression.

Help file generated by VB HelpWriter.

Numeric Bases

See also [Numeric Formats](#)

In addition to the default **decimal** base, UCALC also supports **binary**, **octal**, and **hexadecimal** bases. These numbers must be preceded by the character **#** followed by **b**, **o**, or **h** for binary, octal, or hexadecimal.

Examples:

```
#b110101110  
#h1AE  
#o656
```

The above three numbers are all the same as decimal number 430.

In order for the output to be displayed in one or several of these bases in the *expression evaluator*, select the appropriate options at the bottom of the program.

Help file generated by VB HelpWriter.

Angle Modes

See also [Numeric Bases](#)

The following modes affect trigonometric functions, such as SIN, COS, TAN, SEC, CSC, COT, ASIN, ACOS, and ATAN.

Radian. In this mode one full revolution is 2π radians. 1π radian = 180 degrees. Remember to use this mode when plotting trigonometric functions.

Degree. In this mode one full revolution is divided into 360 equal parts.

Gradient. In this mode one full revolution is divided into 400 equal parts.

Angle Modes can be changed at the bottom, where it says **Angle Mode**.

Help file generated by VB HelpWriter.

Special Copy

UCALC allows editing using standard Windows procedures. You can *Cut*, *Copy*, or *Paste* items by clicking the *right* mouse button, or by selecting options from the *Edit* menu, or by using shortcut keys such as Ctrl-X to cut, Ctrl-C to copy, or Ctrl-V to paste.

In addition to the standard, UCALC also has a *Special Copy* feature which allows you to copy entire tables, such as the *general ledger* , or graphs, or a neatly tabulated list of items such as conversions.

Help file generated by VB HelpWriter.

Contacting the Author

See also [Registration](#)

In addition to intuition and careful analysis of a large number of other programs, much of the way UCALC currently operates is a direct result of suggestions from a number of users like you. If there is a special feature you would like to see in a future version, don't just wish for it to be implemented, but send me a message to let me know about it. Whether you are a professional or a beginner, all suggestions are highly valued. Here is how to contact the author, **Daniel Corbier**:

E-mail

info@ucalc.com (Internet)

Dancorbier@aol.com

75541,1523 (Compuserve)

* E-mail is by far the preferred mode of communication with the author (if it requires a response).

Fax: 305-233-2604

Postal

Daniel Corbier
20410 SW 92 Place
Miami FL 33189, USA

Phone: 305-233-2604 If no one is available at the time of your call, please leave a message including your name and phone number. Also indicate if you are willing to accept a collect call.

WWW

You are invited to visit my web site from time to time in order to obtain the latest information concerning UCALC at <http://www.ucalc.com>.

Help file generated by VB HelpWriter.

Registration Form

See also [Contacting the Author](#)

UCALC 4.0 Registration Form

PSL Part #10655, CIS registration ID 14911. Date _____

Name _____

[Company] _____

[Title] _____

Address _____

City _____ State/Country _____

Zip _____ [E-mail address] _____

[Phone] _____ [Fax] _____

Registration price \$40 X _____ (number of copies) Total _____

(Contact the author, or check www.ucalc.com for site license discounts)

Florida residents add 6 percent sales tax _____

TOTAL ORDER _____

Payment method: Visa Discover Master Card Check
 Cash Money order American Express

Card #: _____ Exp: ___/___ Signature: _____

(Do not send credit card numbers to my home address. Send it to PSL.)

Receive registration code via: E-mail Fax Letter Diskette

--- O / _____
O \ _____ (print the above page and send it with your payment)

(Compuserve users can register online with GO SWREG. Choose ID 14911)

Check <http://www.ucalc.com> for alternate methods of payment.

Credit card registrations are handled by PSL (Please note that I cannot be reached through PSL). Here's how to reach them:

-- Phone PSL at:

1-800-242-4775 (Toll free in the US) To register by credit card.

1-713-524-6394 To register by credit card and for order inquiries.

-- FAX PSL at 713-524-6398

-- Email PSL at CompuServe userid 71355,470

-- Write PSL at:

The Public (Software) Library
P.O. Box 35705
Houston, TX 77235-5705
USA

If you are outside of the US, it is preferable to use a credit card, or international money order. If you use a foreign check, then any surcharges resulting from funds drawn from a non USA bank must be prepaid in addition to the registration cost.

I may accept cash payments at my home address from foreign users who check with me ahead of time for confirmation.

Credit card processing is done by PSL. At my address, I can only accept US postal money orders, international money orders, and checks (and cash when confirmed ahead of time).

Help file generated by VB HelpWriter.

History

UCALC v4.0 (6/4/97)

- Speed improvements for graphs, conversions, and auto session saving
- Reorganized *Options* menu
- Better graph printing
- Zoom out for graphs with *right* click button
- Many small fixes

This is the first Windows version ready for mass distribution.

UCALC v3.8 (3/24/97)

This is the first official Windows release of UCALC following earlier DOS versions. It incorporates major features that are not found in the DOS versions, such as the General Ledger, Unit Converter, and User Solution Modules. It was released on a deadline, and uploaded by the author to only a few places.

UCALC (DOS) versions 1.2 to 3.2 (10/25/92 to 11/21/96)

The earlier versions ran under DOS and included an expression evaluator with graphing capabilities. UCALC for DOS will probably continue to be supported, especially for the benefit of people who use DOS based systems, such as certain palmtops.

Help file generated by VB HelpWriter.

