powerOne® Finance

Version 5.1

www.infinitysw.com

2/4/2005

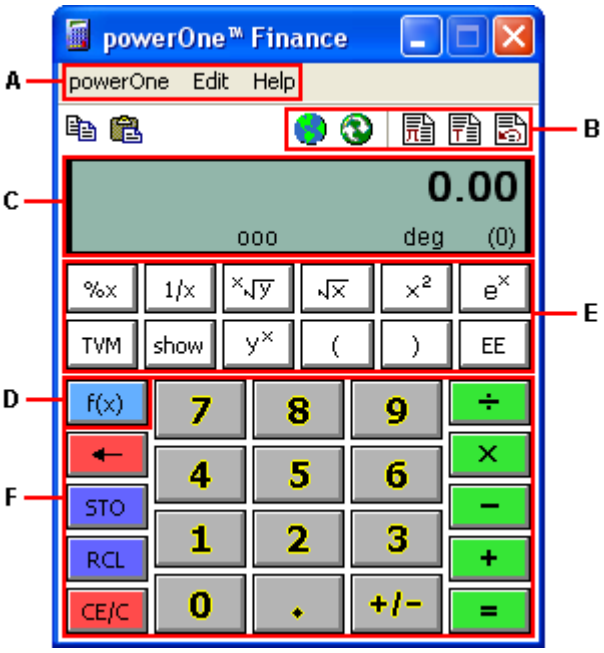
Table of Contents

1 Using the Calculator

1.1 Interface Overview

This section discusses the calculator's main and pop-up interface.

1.1.1 Display



A. Menu (and/or powerOne Button on Palm OS):

- Copy: copy contents of view window to the system clipboard. See Memory & Storage : System Clipboard for more information.
- Paste: paste the system clipboard to the view window. See Memory & Storage : System Clipboard for more information.
- Calculation Log: log of calculations similar to a tape. The Palm OS and Pocket PC versions record the last 20 calculations. See Memory & Storage : Calculation Log for more information.
- Preferences / Options: calculator preferences. See Using the Calculator : Preferences for more information.
- Clear Memory: clears the calculator memory locations.
- Skins: change the user interface of the calculator (colors and layout). See Using the Calculator : Skins for more information. (PalmOS only)
- My Templates: location to see all templates, whether created or pre-installed.
- Contents: application help. (Windows only) For Pocket PC, application help is available via the Start menu. For Palm OS, select (i) in the top, right-hand corner of each template.
- About powerOne: information about the product.

B. Navigation/Synchronization Buttons (from left to right):

- Data Service: select to access updated data. Requires an add-on service. See the Memory & Storage : Data Update Service section for more information.
- Pocket PC Synchronization: select to synchronize data and templates between Pocket PC and Windows version. Only visible in the Windows version of the software if Pocket PC synchronization is available. See the Templates : Synchronizing Templates section or Memory & Storage : My Data : Synchronizing Data section for more information.
- Data Button: displays My Data. See the Memory & Storage : My Data section for more information.
- Template Button: displays list of available templates similar to My Templates. See the Templates : Template List section for more information.
- Last Template Button: select to go to the previously used template (only visible when a template has been visited).

C. View Window: displays calculation and status information. See the Using the Calculator : Input Modes section for more information.

D. Function Button: select this button to display a list of function categories. Select a function category to access a mathematical function.

E. Programmable Buttons: programmable buttons/bar can be changed to any mathematics function or template. Buttons can be changed in the Preferences. See the Using the Calculator : Preferences section for more information.

F. Keypad: calculator keypad consists of numbers, basic arithmetic, backspace, clear, positive/negative button and memory buttons (recall and store).

- 0-9: numbers 0 through 9.
- decimal separator: separate the whole and decimal portions of the number. Either entered as a period or comma depending on the system setting for number display format.
- +, –, x, ÷ (plus, minus, times, divide): basic mathematics functions.
- ENT or equals: enter key to evaluate the equation (algebraic input mode), push a value on the stack (RPN input mode), or complete a calculation (order of operations and chain input modes).
- CE/C: clears the currently entered value on the first selection and all values (entire calculation or history depending on the input mode) on the second selection.
- STO: select to access store memory location functionality. See the Memory & Storage : Memory Locations section for more information.
- RCL: select to access recall memory location functionality. See the Memory & Storage : Memory Locations section for more information.
- +/–: select to change the sign or insert a negative sign depending on the input mode.
- ← (backspace arrow): deletes the highlighted area, space before the input cursor, or last entered value depending on the input mode.

1.1.2 Skins

Skins add a personalized look to the main and pop-up calculators. Skin are only available for the Palm OS version. Each skin can offer a different button layout, advanced mathematics functions, or different color schemes.

To download free skins, go to this product's web page at www.infinitysw.com/finance.

Installing Skins

After downloading a skin from Infinity Softworks' web site and synchronizing it to your device's main memory, run the application. The skin will be imported automatically. To install a skin from an expansion card, select "Skins" from the "powerOne" button and choose "Import" to find it.

Changing Skins

To change skins, select "Skins" from the "powerOne" button. Choose the desired skin and then select "OK". The calculator display will change automatically. "<Default>" is the original display that came with your product.

Deleting Skins

To delete a skin, select "Skins" from the "powerOne" button. Choose the desired skin and select "Delete". The default skin cannot be deleted.

Problems with Skins

If there is a device problem when working in a skin, it is possible to return to the default skin when launching the software. To do so, hold the down scroll or 5-way navigation button when starting the software.

1.1.3 Menus

Palm OS Menu:

Choosing the menu button to the lower, left-hand corner of the Graffiti input area accesses the menus. Standard PalmOS edit choices, Graffiti help, Preferences, and application information can be accessed from here.

The Edit menu:

- **Copy:** shortcut C, copy the selected text to the clipboard.
- **Paste:** shortcut P, paste the selected text from the clipboard to the entry line.
- **Graffiti Help:** shortcut G, help with Graffiti keystrokes.

The Options menu:

- **Preferences:** shortcut R, displays the calculator preferences.
- **Clear Memory:** shortcut Y, clears the calculator's memory locations.
- **About powerOne:** displays company information.

Copy, paste, preferences and the about screen can all be reached from the powerOne button as well.

Pocket PC Menu:

The powerOne menu:

- **Options:** displays the calculator preferences.
- **Calculation Log:** shows the calculation log.
- **My Data**: displays My Data.
- **My Templates**: displays My Templates.
- **Clear Memory:** clears the calculator's memory locations.
- **Exit:** exits the program.

The Edit menu:

- **Copy:** copy the selected text to the clipboard.
- **Paste:** paste the selected text from the clipboard to the entry line.

The About menu:

- displays company information.

Windows Menu:

The powerOne menu:

- **Options:** displays the calculator preferences.
- **Calculation Log:** control-L, shows the calculation log.
- **My Data**: control-D, displays My Data.
- **My Templates**: control-T, displays My Templates.
- **Clear Memory:** clears the calculator's memory locations.
- **Exit:** exits the program.

The Edit menu:

- **Copy:** control-C, copy the selected text to the clipboard.
- **Paste:** control-V, paste the selected text from the clipboard to the entry line.

The Help menu:

- **Contents:** displays the manual.
- **About:** displays company information.

1.1.4 Pop-up Calculator

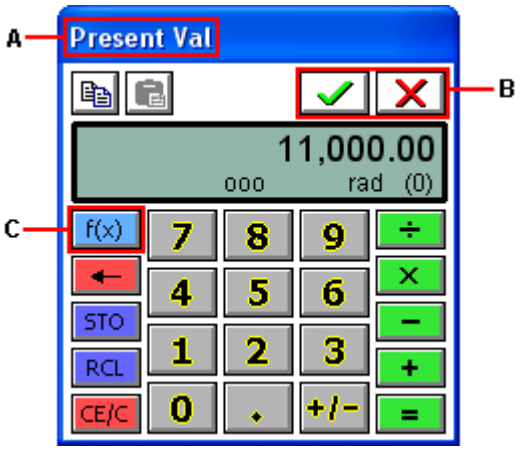
The pop-up calculator is used throughout the application when values are required, such as in a table or template. The pop-up calculator functions similarly to the main calculator and offers the same input modes. Functionality specific to the pop-up calculator is detailed here. See the Interface Overview : Display section for information on shared main and pop-up calculator functionality and the Input Modes section for information specific to each available input mode.

A. Variable Name: displays the name of the selected variable.

B. Buttons (from left to right)**:**

- Save Button: select the "✓" button to store the value in the view window and return to the previous view.
- Cancel Button: select the "x" button to return without storing.

C. Function Button: displays a list of functions available in the pop-up calculator. This list's functionality depends on the currently selected input mode. See the Using the Calculator : Input Modes for more information.



In general, entries made in the pop-up calculator are separate from those in the main calculator. To move data between them, store the information in a memory location (MEM : Store). See the Using the Calculator : Memory & Storage section for more information.

1.2 Input Modes

An input mode is the method by which calculations are performed. The different modes reflect the variety of calculation methods performed by currently available hardware calculators.

The currently selected skin dictates the available input modes. See Using the Calculator : Interface Overview : Skins for more information.

The input mode is set in the preferences. See Using the Calculator : Interface Overview for more information on accessing the preferences.

1.2.1 RPN Mode

This is the input mode used by financial calculators and some engineering calculators. RPN input mode uses a chain of values and then a chain of operations to perform the computations. This mode utilizes a stack, which stores numerical entries. The stack works like a pile of dishes. Entering a number is like putting a plate on top of the pile. This is called pushing onto the stack. To push a variable, enter the number then press "ENT" (Enter). Performing a calculation is like taking a plate off the pile of dishes. This is called popping off of the stack. To do this, press a two-variable function.

When a calculation is performed, the number at the top of the stack is the first operand while the number in the visible view window is the second. The stack can be viewed at any time by selecting the "stack" function. See the section on RPN Stack and History List for more details.

To calculate 27 + 3 x 8.5:

Key	Display	Comments
C/CE		Tap twice to clear the display
27	27	
ENT	27.00	
3	3	
ENT	3.00	
+	30.00	
8.5	8.5	
ENT	8.50	
x	255.00	

The following keys can be used to interact with the calculator.

Character	Function	Character	Function
0	Zero	<back>	Backspace
1	One	c or <back>	C/CE
2	Two	+	Add
3	Three	–	Subtract
4	Four	x or *	Multiply
5	Five	/	Divide
6	Six	ENT or <return>	Enter
7	Seven	s (Windows only)	Store to memory
8	Eight	r (Windows only)	Recall from memory
9	Nine	f (Windows only)	Shows function list
. or ,	Decimal Pt		
n	Sign		
e	Exponent		

1.2.1.1 Display

A. View Window: number display area.

B. Status Indicators:

- Shift Indicator: Standard Palm OS Graffiti shift indicator. (Palm OS only)
- Clear: Clear serves two functions in this mode. Selecting it once displays the clear indicator and clears only the number currently being entered. Selecting it the second time clears the entire calculation. If the status indicator "clear" appears, the currently entered number has been cleared.
- Deg/Rad: Whether calculator is in degrees or radians mode.
- RPN: Currently in order of operations input mode.

1.2.1.2 Stack

There are special functions for manipulating the stack. These functions can be reached by choosing the category Stack from the function list and then selecting the desired function. If an item is selected on the stack, the function selected will adjust based on that item. If a stack function is selected from the function list the first item on the stack will be the focal point.

- **Drop:** deletes the item.
- **Duplicate** (dup)**:** copies the item into register 0 (view window/entry line), pushing all others up.
- **Move:** removes the item from its location in the stack and places it in register 0 (view window/entry line).
- **Rotate** (rot)**:** moves the stack in a clockwise direction.
- **Rotate Rvrs** (rotr)**:** moves the stack in a reverse or counter-clockwise direction.
- **Swap:** swaps the item with the contents of register 0 (view window/entry line).

1.2.1.3 Preferences

The following preferences are available in RPN input mode. See the Using the Calculator : Preferences section for more information:

- **Stack Size:** 4 or 11 registers. Includes register 0 (view window/entry line).
- **Decimal Setting:** float or 0 through 11.
- **Display Mode:** normal, scientific or engineering notation.
- **Trig Mode:** degrees or radians.

1.2.1.4 Functions

Different input modes offer different functions. This lists the available functions for RPN input mode. To learn how to see a complete list of functions organized by category, see the Using the Calculator : Interface Overview section. To learn more about an individual function, see the Functions section.

math		
1/x	log	mod
%x	10^x	gcd
y^x	x^2	lcm
ln	$\sqrt{x}$	
e^x	$\sqrt[x]{y}$	

number		
last	iPart	sign
show	fPart	round
EE	floor	degs
abs	ceil	dms

trig		
sin	sinh	sec
cos	cosh	csc
tan	tanh	cot
asin	asinh	degrees
acos	acosh	radians
atan	atanh	

prob
nPr
nCr
x!
rand

cnst
e
pi

stack
drop
dup
move
rot
rotr
swap
stack

1.2.2 Order of Operations Input Mode

Order of operations is a standard calculator mode where only the currently entered number appears in the view window. This mode follows standard entry where numbers are entered in order of appearance and a final calculation is performed when [=] is selected. As the name indicates, it follows standard order of operations rules.

The following chart outlines precedence:

Order	#Function
1	Negative (-x)
2	Powers and roots
3	Multiplication, division, and percentages
4	Addition and subtraction
5	Relationship operations (>, <=)
6	Logic or Boolean operations (or, and)

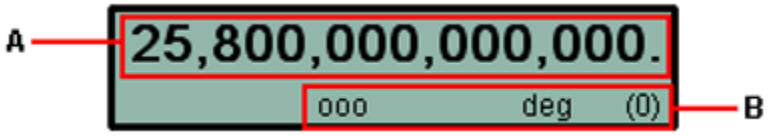
To calculate 27 + 3 x 8.5:

Key	Display	Comments
CE/C		Clears the current calculation.
27	27	
+	27	
3	3	
x	3	
8.5	8.5	
=	52.5	

The following keys can be used to interact with the calculator.

Character	Function	Character	Function
0	Zero	<back>	Backspace
1	One	c or <back>	C/CE
2	Two	+	Add
3	Three	–	Subtract
4	Four	x or *	Multiply
5	Five	/	Divide
6	Six	= or <return>	Equals/Enter
7	Seven	(	Lt Paren
8	Eight	)	Rt Paren
9	Nine	s (Windows only)	Store
. or ,	Decimal Pt	r (Windows only)	Recall
n	Sign	f (Windows only)	Shows function list
e	Exponent		

1.2.2.1 Display



A. View Window: number display area.

B. Status Indicators:

- Shift Indicator: Standard Palm OS Graffiti shift indicator. (Palm OS only)
- Clear: Clear serves two functions in this mode. Selecting it once displays the clear indicator and clears only the number currently being entered. Selecting it the second time clears the entire calculation. If the status indicator "clear" appears, the currently entered number has been cleared.
- Deg/Rad: Whether calculator is in degrees or radians mode.
- OOO: Currently in order of operations input mode.
- (0): parentheses indicator. The number in the middle shows the number of left parentheses that are still open (i.e., do not have a closing right parenthesis).

1.2.2.2 Preferences

The following preferences are available in order of operations input mode. See the Using the Calculator : Preferences section for more information:

- **Decimal Setting:** float or 0 through 11.
- **Display Mode:** normal, scientific or engineering notation.
- **Trig Mode:** degrees or radians.

1.2.2.3 Functions

Different input modes offer different functions. This lists the available functions for order of operations input mode. To learn how to see a complete list of functions organized by category, see the Using the Calculator : Interface Overview section. To learn more about an individual function, see the Functions section.

math		
1/x	log	mod
%x	10^x	gcd
y^x	x^2	lcm
ln	√x	
e^x	$\sqrt[x]{y}$	

number		
last	abs	sign
show	iPart	round
(	fPart	degs
)	floor	dms
EE	ceil	

trig		
sin	sinh	sec
cos	cosh	csc
tan	tanh	cot
asin	asinh	degrees
acos	acosh	radians
atan	atanh	

prob
nPr
nCr
!x
rand

cnst
e
pi

history *

* History is a special function that displays the last 10 recorded answers. Each time the equals [=] button is selected, a

new history item is added to the list. To recall a history item to the view window, select it from the list (or choose function Last for the last value added to the history list).

1.2.3 Chain Input Mode

Chain input mode is the most common mode found in financial calculators. It is a mode where only the currently entered number appears in the view window. This mode follows standard entry where numbers are entered in order of appearance and a final calculation is performed when [=] is selected. Order of operations is ignored, opting instead to evaluate as numbers are entered and operands are selected.

To calculate 27 + 3 x 8.5:

Key	Display	Comments
CE/C		Clears the current calculation.
27	27	
+	27	
3	3	
x	30	
8.5	8.5	
=	255	

The following keys can be used to interact with the calculator.

Character	Function	Character	Function
0	Zero	<back>	Backspace
1	One	c or <back>	C/CE
2	Two	+	Add
3	Three	–	Subtract
4	Four	x or *	Multiply
5	Five	/	Divide
6	Six	= or <return>	Equals/Enter
7	Seven	(	Lt Paren
8	Eight	)	Rt Paren
9	Nine	s (Windows only)	Store
. or ,	Decimal Pt	r (Windows only)	Recall
n	Sign	f (Windows only)	Shows function list
e	Exponent		

1.2.3.1 Display

A. View Window: number display area.

B. Status Indicators:

- Shift Indicator: Standard Palm OS Graffiti shift indicator. (Palm OS only)
- Clear: Clear serves two functions in this mode. Selecting it once displays the clear indicator and clears only the number currently being entered. Selecting it the second time clears the entire calculation. If the status indicator "clear" appears, the currently entered number has been cleared.
- Deg/Rad: Whether calculator is in degrees or radians mode.
- Chain: Currently in chain input mode.
- (0): parentheses indicator. The number in the middle shows the number of left parentheses that are still open (i.e., do not have a closing right parenthesis).

1.2.3.2 Preferences

The following preferences are available in chain input mode. See the Using the Calculator : Preferences section for more information:

- **Decimal Setting:** float or 0 through 11.
- **Display Mode:** normal, scientific or engineering notation.
- **Trig Mode:** degrees or radians.

1.2.3.3 Functions

Different input modes offer different functions. This lists the available functions for chain input mode. To learn how to see a complete list of functions organized by category, see the Using the Calculator : Interface Overview section. To learn more about an individual function, see the Functions section.

math		
1/x	log	mod
%x	10^x	gcd
y^x	x^2	lcm
ln	√x	
e^x	$\sqrt[x]{y}$	

number		
last	abs	sign
show	iPart	round
(	fPart	degs
)	floor	dms
EE	ceil	

trig		
sin	sinh	sec
cos	cosh	csc
tan	tanh	degrees
asin	asinh	radians
acos	acosh	
atan	atanh	

prob
nPr
nCr
!x
rand

cnst
e
pi

history *

* History is a special function that displays the last 10 recorded answers. Each time the equals [=] button is selected, a new history item is added to the list. To recall a history item to the view window, select it from the list (or choose function Last for the last value added to the history list).

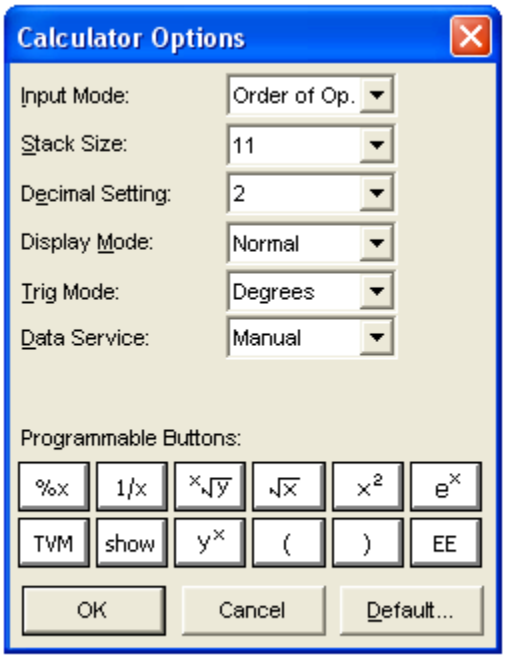
1.3 Preferences

Preferences (or Options) are used to store information about how the calculator functions and set the programmable buttons. The "Button" tabs only appear if the selected skin offers these functions.

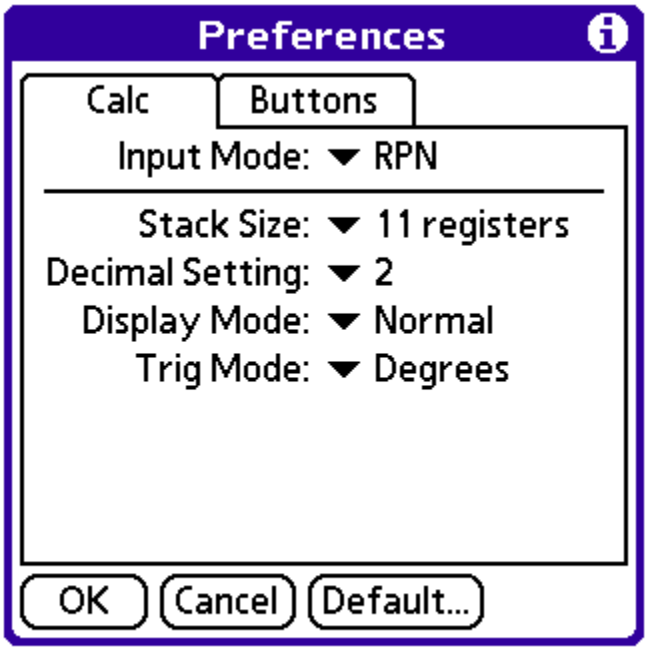
Access the preferences by selecting "powerOne" then "Preferences" on Palm OS devices or "File" then "Options" on Pocket PC and Windows computers.

1.3.1 Calc Tab

Display changes depending on the input mode selected. Options are:



Windows & Pocket PC



Palm OS

Input Mode

[all modes] the currently selected input mode.

All input modes may not be available with all skins. For more on each input mode, see the Using the Calculator : Input Modes section. To learn more about skins, see the Using the Calculator : Interface Overview : Skins section.

Stack Size

[RPN] size of the stack.

The stack can either be 4 items tall (3 locations and the entry line/view window) or 11 items (10 locations and the entry line/view window).

Decimal Setting

[RPN, order of operations, chain modes] number of decimal places to display.

Float shows all available decimal places. 0 through 11 shows that many decimal places. With very large numbers, fewer decimal places may be displayed because of the total number of places available to show in the view window. In addition, the Show function displays all available decimal places until the next entry is made. This can be used to quickly see all available decimal places when the decimal setting is not set to float.

Display Mode

[RPN, order of operations, chain modes] display numbers in normal, scientific or engineering notation.

Normal mode displays numbers as would normally be written on paper or, if the number is very large or very small, in scientific notation. Scientific mode displays numbers as two parts -- the significant digits with one digit before the decimal and the exponent (e.g, 3.45e67). The number of places displayed after the decimal point is determined by the decimal setting. Engineering mode also displays numbers in two parts -- the significant digits and the exponent that is always a multiple of 3 (e.g, 34.567e12). The number of significant digits is 1 plus the decimal setting.

Trig Mode

[RPN, order of operations, chain modes] calculates trigonometric functions as either degrees or radians.

Data Service

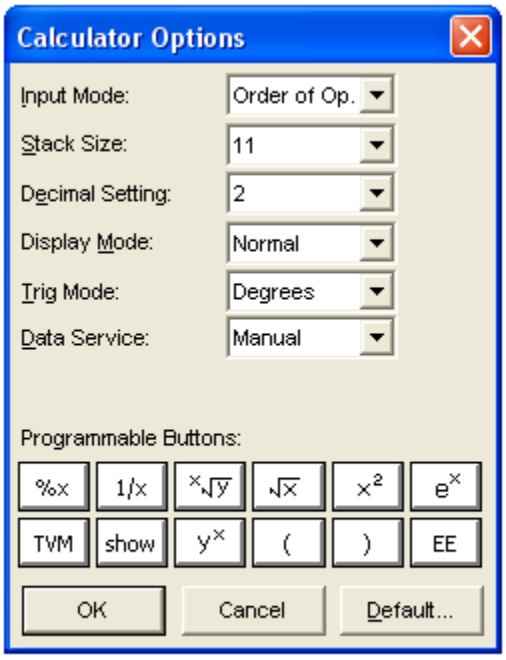
[RPN, order of operations, chain modes] available in the Windows version only, designates how often the Data Update Service should update data. Options include:

- Manual: updates only when the Data Update Service navigation button is selected.
- Auto: 1hr: updates automatically when the application is started and every one hour thereafter if the application remains running.
- Auto: 3hr: updates automatically when the application is started and every three hours thereafter if the application remains running.
- Auto: 6hr: updates automatically when the application is started and every six hours thereafter if the application remains running.
- Auto: 12hr: updates automatically when the application is started and every 12 hours thereafter if the application remains running.

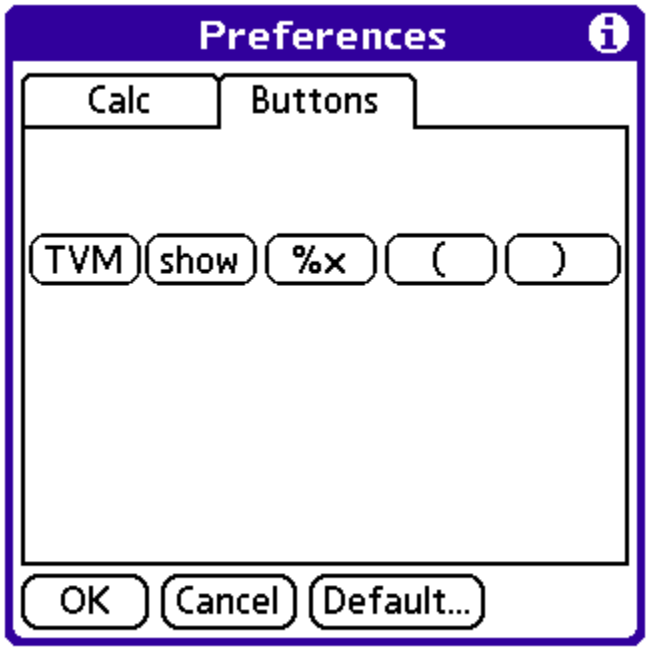
See the Memory & Storage : Data Update Service section for more information.

1.3.2 Button Tab

On Windows and Pocket PC operating systems, the buttons are in the same window as the rest of the preferences. On the Palm OS, the buttons are found by selecting the button tab.



Windows & Pocket PC



Palm OS

To change the content of any button, select it and choose a new item from the list.

On the Palm OS version, choose a function by selecting its category and then the function itself. At the bottom of the list are Category, Calc Log, Template and Empty. Select Category to set the button to a Function Category. Select Calc Log to set a button to the Calculation Log. Select Template to set an individual template to a button (requires a shortened name). To leave the button blank, select Empty.

On the Windows and Pocket PC versions, choose a function by selecting it from the list. At the bottom of the list are clog and templates. Select clog to set a button to the Calculation Log. Select Template to set an individual template to a button

(requires a shortened name).

Programmable buttons appear like other buttons in the main calculator.

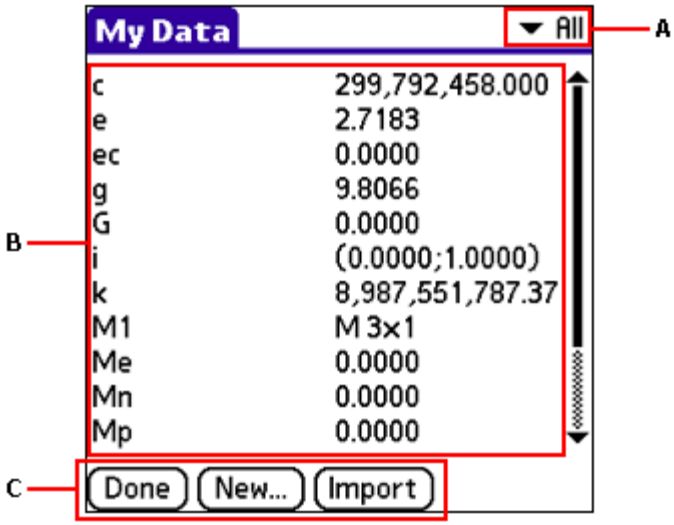
1.4 Memory & Storage

Storing data is important when performing advanced calculations. Multiple methods are available for storing and moving data within this product and with other, external sources. Most are outlined in this section.

1.4.1 My Data

My Data displays created data, whether functions, variables or constants. Stored data can be variables, macros, constants, tables or matrices. See the Using the Calculator : Interface Overview section for more information on accessing My Data.

A. Categories: select the category name to choose a different display. "All" shows all data at the same time, whether variables, functions or constants. Other categories show just their corresponding data type.



B. Data Listing: each data line shows the variable's name to the left and the value of that variable to the right. Select a data line to show options:

- Use: select to recall the value to the main calculator. In RPN input mode, this pushes the data item onto the stack. In order of operations and chain input modes, it returns the value. Data items can also be found by selecting the cnst or vars category in the function list.
- Edit: shows the data editor. See the appropriate Types of Data section for more information on creating and editing data.
- Export/Beam: select to show export and beam options for the data item. See the Sharing Data section for more information.
- Delete: select to delete the data item.
- Duplicate: select to make a copy of the data item.
- Notes: display notes pertaining to the selected data item.

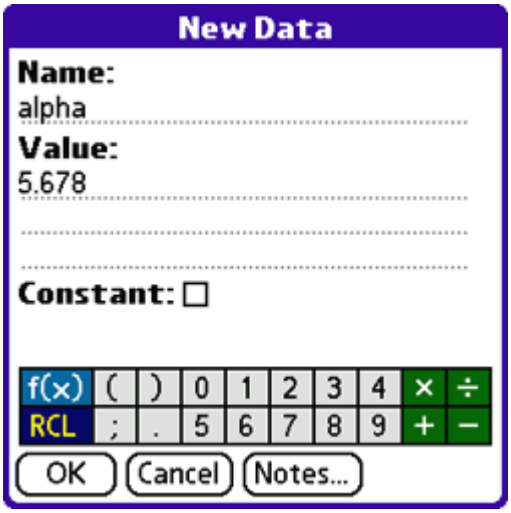
C. Buttons: to create a new data item, select "New" and choose the type of data to create. For more information on each type of data, see its corresponding discussion within this section. To import data, select "Import". See the Sharing Data section for more information. Select "Done" to leave My Data.

1.4.1.1 Variables

Variables are individual values. This section discusses single value variables.

New/Edit Variables from My Data

Generally, variables are entered from My Data. Select "New" for a new variable. Select a variable from the My Data data list and choose "Edit" to edit a variable.



A. Name: variable's name. The name consists of letters (capital or lower case) and optionally numbers 0-9. A name cannot start with a number. Note that the name is case sensitive, meaning that variable "abc" is different from variable "ABC". See the Appendix : Restricted Data Names section for names that should not be used.

B. Value: variable's value. Values can be entered either as a number [e.g., 3.5] or as an expression [e.g., 3.5*5^6]. If entered as an expression, it will be evaluated before storing. For instance, if 3.5*5^6 were entered, that variable would be stored as 54,687.5.

C. Constant: whether the variable is a constant or not. For variables, the Constants checkbox is unchecked.

D. Keypad: keypad for easy entry. "RCL" displays the memory location list. "f(x)" displays the list of functions organized by category. This is the same as the main calculator's list except category MEM is added (memory store, recall and clear options). See the appropriate Input Modes section for more on which functions are available in each mode. See the Memory & Storage : Memory Locations section for more on recall and store.

E. Buttons: "OK" saves changes while "Cancel" deletes changes, returning to My Data. To enter notes about the data item, select "Notes" or enter them in the appropriate field.

1.4.1.2 Constants

Constants are defined variables that cannot be altered. If a new data item is created with the same name as a constant, a warning will appear before saving the new data. A number of constants are included automatically.

New/Edit Constants from My Data

Constants are entered from My Data. Select "New" for a new constant. Select a constant from the My Data data list and choose "Edit" to edit the constant.

A. Name: constant's name. The name consists of letters (capital or lower case) and optionally numbers 0-9. A name cannot start with a number. Note that the name is case sensitive, meaning that constant "abc" is different from constant "ABC". See the Appendix : Restricted Data Names section for names that should not be used.

B. Value: constant's value. Values can be entered either as a number [e.g., 3.5] or as an expression [e.g., 3.5*5^6]. If entered as an expression, it will be evaluated before storing. For instance, if 3.5*5^6 were entered, that variable would be stored as 54,687.5.

C. Constant: type of data to store.

D. Keypad: keypad for easy entry. "RCL" displays the memory location list. "f(x)" displays the list of functions organized by category. This is the same as the main calculator's list except category MEM is added (memory store, recall and clear

options). See the appropriate Input Modes section for more on which functions are available in each mode. See the Memory & Storage : Memory Locations section for more on recall and store.

E. Buttons: "OK" saves changes while "Cancel" throws out changes, returning to My Data. To enter notes about the data item, select "Notes".

Included Constants

Function	Display	Value
Speed of Light	c	299,792,458 m/s
Exponential (cannot be edited)	e	2.71828182846
Elemental Charge	ec	1.60217646E-19 C
Gravity Acceleration	g	9.80665 m/s^2
Gravity Constant	G	6.67259E-11 $m^3/kg\ s^2$
Coulomb	k	8,987,551,787.37
Electron Mass	Me	9.10938188E-31 kg
Proton Mass	Mp	1.67262158E-27 kg
Neutron Mass	Mn	1.67492716E-27 kg
Avogadro's Number	NA	6.02214199E23/mol
Pi (cannot be edited)	pi	3.14159265359
Universal Gas Constant	R	8.314472 J/mol K

* not available in order of operations and chain input modes.

The constant Tolerance is also available. Tolerance is used for derivative, second derivative, function maximum and function minimum when the tolerance is not included in the mathematical function (if available). It defaults to 0.0001

1.4.1.3 Sharing Data

This section discusses sharing data.

Export/Beam
Palm OS: export/beam data.

- Select the data to share.
- Choose "Export/Beam" from the list if the data can be shared. If it cannot be shared, this option does not appear. Export/Beam Options appears.
- Select the desired export/beam option.
- Follow the on-screen directions, if any are required.

Pocket PC/Windows: export templates or its data (beam not available).

- Select or right-select the data to export.
- Choose "Export/Beam" from the list if the template can be shared. If it cannot be shared, this option does not appear. Export dialog appears.
- Enter a name and location information.
- Select OK or Save to save the file.

The data export/beam options that come with the software:

- **Export data item to file:** save the selected template in a file that can be synchronized to the desktop for archival or sharing purposes.
- **Beam data item:** beam the selected template to another handheld that has this software.

Infinity Softworks may offer additional export/beam plug-ins from its web site (Palm OS only). These plug-ins could include ones to communicate with word processors, spreadsheets and probe systems, among others. See the Plug-ins web page at www.infinitysw.com/finance for more information.

Import
Import templates.

- Select "Import" at the bottom of My Templates.
- Select the desired import option.
- Pick a location and choose, Save or OK, if required.

Data, generally, are imported automatically when the software is first started. On Palm OS handheld computers, any data uploaded to the device will be imported. On Pocket PC and Windows, any data installed to the "powerOne" directory in "My Documents" will be installed automatically. If the data is on an expansion card, CD-ROM or not in the powerOne directory on Pocket PC and Windows computers, use the Import option in My Data to install.

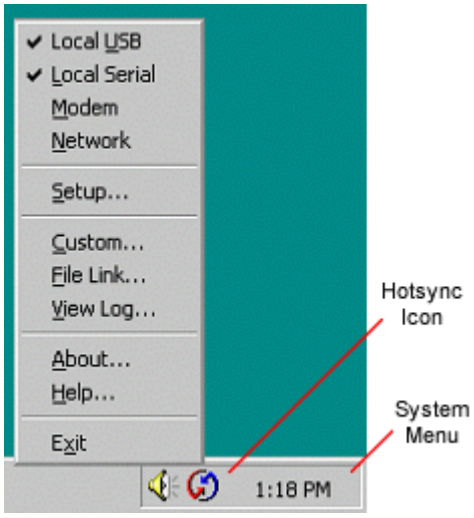
1.4.1.4 Synchronizing Data

If you own both handheld and Windows versions of the software, it is possible to synchronize data and templates.

Palm OS

By default, synchronization happens each time the HotSync button is selected on the cradle or on the device.

For Palm OS devices, synchronization can be turned off, set to copy to the handheld only, to the Windows computer only, or synchronized both directions (two-way sync). With two way synchronization, the more recently changed template will be transferred. To change this preference for the next HotSync operation, tap the HotSync icon in the system tray of your desktop computer (which is usually in the bottom right corner of the screen):



Select "Custom…" from within the "HotSync" menu in your Palm Desktop software. Double-click the powerOne application and change your preference:

Pocket PC

The data is synchronized each time the sync icon is selected in the desktop version of the software. See the Using the Calculator : Interface Overview : Display section for more information.

1.4.2 Data Update Service

Infinity Softworks' powerOne Data Service offers updated data from the Internet. This data, updated multiple times throughout the day, can be used in both the calculator and in the templates.

An example of an item included in powerOne Data Service is currency exchange rates. With this Service, log on multiple times per day to keep your rates up-to-date.

For more information about signing up for the data update service, go to:

www.infinitysw.com/dataservice

1.4.3 Memory Locations

There are 10 memory locations available. Memory locations can be accessed one of two ways, depending on the skin. In some skins, select "MEM" and either "Store" or "Recall". In other skins, select "RCL" or "STO" (or "r" or "s") buttons. See the Using the Calculator : Interface Overview section for information on accessing these locations. The same memory locations are available in the main and pop-up calculators.

When storing, the contents of the entry line/view window will be stored to the selected memory location. Only the visible value in the view window is stored.

When recalling, the value will be placed in the entry line at the current cursor position. When recalling in order of operations and chain, the value will overwrite the contents of the view window.

It is possible to perform arithmetic operations on a value that is being stored to memory using existing values in memory. This allows values from the entry line/view window to be added to or subtracted from a current value in memory, as well as several other operations. Note that operations performed on memory locations do not affect the contents of the entry line.

The following table describes the operations that can be performed on memory locations, where 'value' refers to the contents of the entry line/view window while memory location means the contents of a selected memory location. Calculations are performed as follows:

Operand	Functionality
=	Overwrites the selected memory location with the value.
+	The memory location plus the value
–	The memory location minus the value
x	The memory location multiplied by the value
÷	The memory location divided by the value
y^x	The memory location to the value's power.

For example, to add the last calculated result to memory location 2:

- select "MEM" then "Store" (or the "STO" button, depending on the skin in use)
- tap "+"
- tap memory location 2

Clearing memory locations also depends on the skin. If the skin has a "MEM" button, select this button and choose "Clear". Alternatively, select "Clear Memory" from the Options menu. See the Using the Calculator : Interface Overview :

Menus section for more information.

1.4.4 System Clipboard

The system clipboard can be used to store values or move information from one application to another.

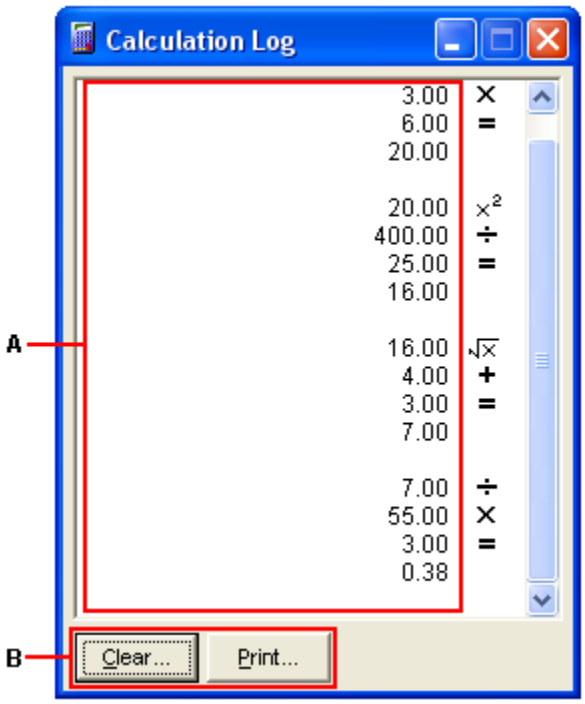
Select an appropriate function from the Edit menu (or Copy or Paste by selecting the "powerOne" button). The view window's value will be the target.

1.4.5 Calculation Log

The calculation log is used to display computations as they are entered. Even if the calculator input is cleared, the calculation log retains a history of the last calculations. To access the calculation log, see the Using the Calculator : Interface Overview.

A. Value: selecting the value returns it to the entry line/view window for use in the calculation.

B. Additional Functionality: in all modes, select "Clear" to clear the calculation log, or select "Print" to print the entire calculation log. Print is available in the Windows version only. Switching input modes also clears the calculation log automatically.



2 Types of Data

When working with data, it is important to understand that data comes in various forms. This section outlines the core data types, how they are used, and what functions are available for each. The Function List section discusses each function in more depth and its relationship to the appropriate data types. In general, many of these data types will only be used when creating templates.

2.1 Booleans

Booleans, only available in modes that use an entry line, are either true or false and are used when comparing two items or logic functions.

See Subject Areas : Booleans for more information.

2.2 Integers (Whole Number)

Integers are whole numbers, numbers without a fractional part. Numbers such as 3, -45, and 0 are integers. Integer numbers are generally in the range –2,147,483,648 to 2,147,483,647 or -2^{31} to $2^{31}-1$.

2.3 Floating Point Numbers

Floating point numbers are numbers with a fractional portion. Numbers such as 3.5, -75.235 and 1E-2 are floating point numbers. The number 3.0 is also a floating point number because of the .0 portion -- if it were written as 3 it would be an integer number. Floating point numbers are in the range 1E308 to -1E308.

If the input mode allows, the base-fraction mode should be set to decimal, fraction or mixed fraction. In decimal mode, answers are returned as floating point numbers or integers. In fraction and mixed fraction mode, answers are returned as fractions. In other modes, the entries are rounded before proceeding. See the appropriate section in Using the Calculator : Input Modes for more information on base-fraction modes.

2.4 Dates & Times

Dates, only available in modes that use an entry line, refer to both month-day-year and time of day. There are two formats:

- **Date:** day-month-year entered as dd.mmyyyy, a 2-digit day, 2-digit month and 4-digit year. Years must be between 1900 and 3000. An example is September 27, 1987. This would be entered as "27.091987" (without quotation marks).
- **Time:** hour-minute-second-millisecond entered in hh.mmssmmm format, 2-digits each for hour, minute and second and 3-digits for millisecond. Hours should be entered in military time (0-23 hours). For example, 11:05am would be entered as "11.05" while 11:05 pm would be entered as "23.05" (without quotation marks in both examples).

See Subject Areas : Dates & Times for more information.

3 Subject Areas

3.1 Boolean

See Types of Data : Boolean for a detailed definition.

Boolean values can be used in many functions and operands that uses integers. The following functions are commonly

used specifically with Boolean values (category in bold):

bool	
==	!x
<>	&&
<	||
>	##
<=	if
>=	choose

Examples:

3.5 < 5 returns true

3.5 > 5 returns false

if (5 <= x; 25; 75) returns 25 if 5 is less than or equal to x, otherwise it returns 75

if (x > 3 && x < 15; 0; 20) returns 0 if x is greater than 3 and less than 15, otherwise it returns 20

See the Functions section for additional information on each function.

3.2 Dates & Times

See Types of Data : Date for a detailed definition.

The following functions are commonly used with date and time mathematics (category in bold):

date	
adjDate	getDate
adjTime	getTime
makeDate	HRS
wkDay	HMS
dDays	today

See the Functions section for additional information on each function.

The following templates are commonly used with data and time calculations (category in bold):

Calendar
Date
Time

See the Templates : Included Templates section for additional information on each template. See Infinity Softworks' web site for additional templates: www.infinitysw.com/finance

3.3 Trigonometry

The following functions are commonly used specifically with trigonometric mathematics (category in bold):

trig					
sin	asin	sinh	asinh	sec	degrees
cos	acos	cosh	acosh	csc	radians
tan	atan	tanh	atanh	cot	

See the Functions section for additional information on each function.

4 Templates

4.1 Accessing

There are three ways to get to a template:

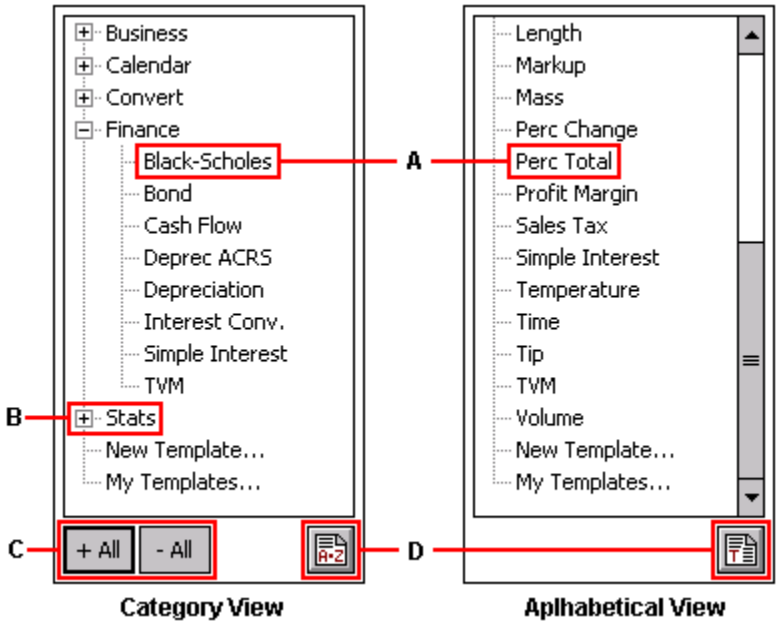
1. In the main calculator, select the Template button at the top of the screen to go see a list of templates. See the Using the Calculator : Interface Overview section for more on this button.

2. In the main calculator, select the Last Template button at the top of the screen to go to the last opened template. See the Using the Calculator : Interface Overview section for more on this button.

3. In the main calculator, select "powerOne" then select "My Templates".

4.2 Template List

The Template List displays a list of available templates and their categories. This list can be viewed in category order or in alphabetical order:



A. Template: select a template to go to that template.

B. Template Category: templates are organized into categories. If in category view and the category is closed (the plus symbol is visible next to the category), select it to open the category and see that category's templates. If in category view and the category is open (the minus symbol is visible next to the category), select it to close the category and hide that category's templates. If there are no templates in the category, that category will not be visible.

C. Open/Close All Buttons: select a button to either open all categories or close all categories.

- + All: select to open all categories.
- – All: select to close all categories.

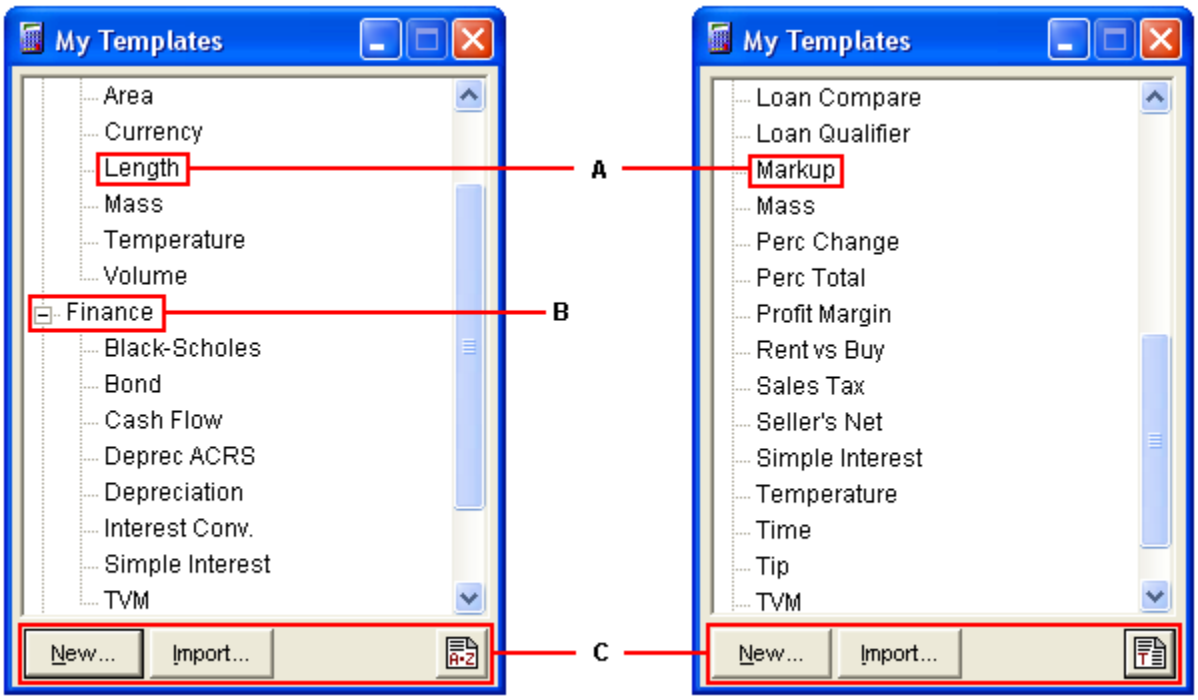
D. Switch View Button: select to switch between category and alphabetical views.

E. Additional Options: at the bottom of the list in both category and alphabetical view are three additional options: to create a new template or go to My Templates.

- New Template: select to create a new template using the solver. See the Creating Templates section for more information.
- My Templates: select to go to My Templates. See the My Templates section for more information.

4.3 My Templates

My Templates displays a list of available templates and their categories. This list can be viewed in category order or in alphabetical order:



Category View **Alphabetical View**

A. Template: select the template for a list of options.

- Use: select to display the template for use.
- Edit: select to edit the template's equation. Only visible if the template can be edited.
- Export/Beam: select to export or beam templates. See the Using the Templates : Sharing Templates & Data section for more information. Only visible if the template can be exported and/or beamed.
- Delete: select to delete the template. Only visible if the template can be deleted.
- Duplicate: select to duplicate the template. Only visible if the template can be edited.
- Notes: select to view the template notes.

B. Template Category: templates are organized into categories. If in category view and the category is closed (the plus symbol is visible next to the category), select it to open the category and see that category's templates. If in category view and the category is open (the minus symbol is visible next to the category), select it to close the category and hide that category's templates. If there are no templates in the category, that category will not be visible.

C. Buttons: (from left to right)

- New: select to create a new template using the solver. See the Templates : Creating Templates section for more information.
- Import: select to import template. See the Using the Templates : Sharing Templates & Data section for more information.
- Switch View: select to switch between category and alphabetical views.

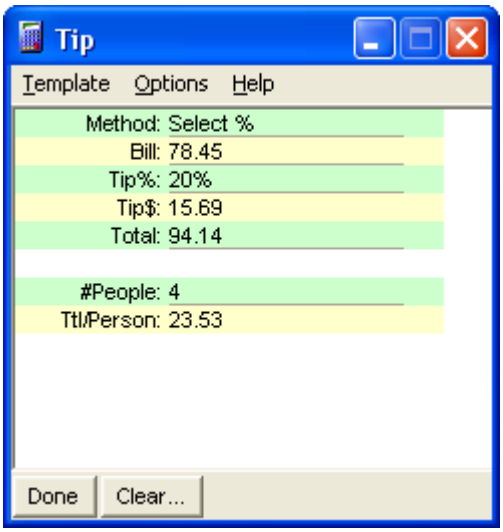
4.4 Using the Templates

This section discusses how templates are generally used. See the Templates : Included Templates section for details on each specific template.

4.4.1 Quick Start Example

The Example

This is an example to quickly demonstrate how to use a template. 4 colleagues go to dinner and spend $78.45. How much is paid for the tip and total bill with a 20% tip? How much was the tip? If the bill were split evenly, what does each colleague have to pay?



Go to the Tip Template

1. Launch the software. The main calculator should be visible.

2. Select the Template button. The Template button is third from the left across the top of the main calculator. The Template List should be visible.

3. Open the Business category by selecting its name.

4. Open the Tip template by selecting its name. The Tip template should be visible.

Calculate Tip Amount and Total Bill

5. Enter $78.45 for the bill.
   - select 0.00 next to Bill.
   - enter 78.45 in the pop-up calculator using the keypad.
   - select the save ("✓") button.

6. Enter a 20% tip.
   - select 15% next to Tip.
   - choose 20% from the list.

7. Calculate the tip amount.
   - select the "?" button on the same line as Tip$.
   - the tip amount is $15.69.

8. Calculate the total bill.
   - select the "?" button on the same line as Total.
   - the total bill is $94.14.

Split the Bill

8. Enter 4 for the number of people.
   - select 1 next to #People.
   - enter 4 in the pop-up calculator using the keypad.
   - select the save ("✓") button.

9. Calculate the total per person.
   - select the "?" button on the same line as Ttl/Person.
   - the total per person is $23.53.

4.4.2 Interface Overview

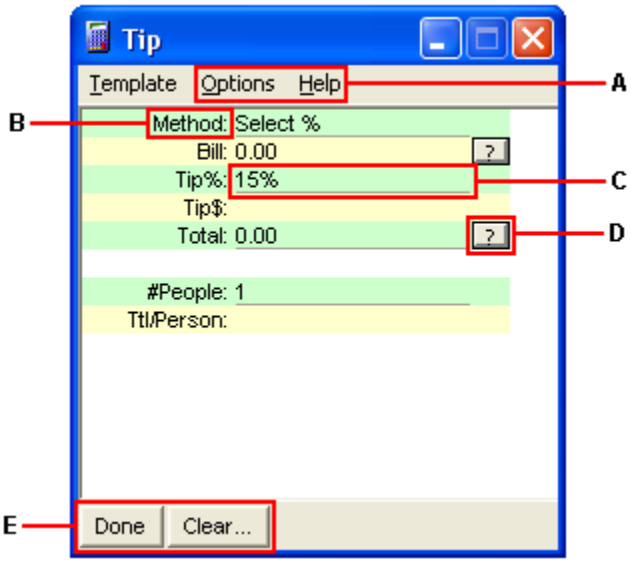
All templates are similar in design. The template itself is broken into 3 columns: the first describes the variable, the second is the variable's data and the third indicates which variables can be calculated. To use any template, enter the known variables by selecting the Data column and entering each. If a table of data is required, choose or create one by selecting the label for that variable. To calculate the unknown variable, select the "?" button on its row.

For details not noted here and examples on how to use individual templates, see the desired template in the Included Templates section.

A. Template Notes: select this button to display notes about the template. Notes generally describe what the template is used for, a description of each variable and any special template instructions.

B. Variable Label: describes the variable's contents. Generally, the label is nothing more than a text description, but sometimes labels can be selected. Labels that can be selected are surrounded by a dotted border and exist for one of two reasons:

- Select a Table: some templates use a table. See the Using the Templates : Types of Variable Data section for more information on selecting and using a table.
- Change the Data View: some variable's can be entered or viewed in multiple ways. For example, the variable Diff H.MM in the Time template. In this case, select the variable's label to enter and calculate the same data item a different way. In the Time template, the difference in time can be entered in hour-minute (Diff H.MM) or fractional hour (Diff Hrs) format.



C. Variable Data: the value or data for the variable. To enter data, select its value and enter it. The data can be entered if it is underlined; otherwise it can only be calculated. Data can be shown in multiple ways. See the Using the Templates : Types of Variable Data section for more information.

D. Calculate "?": indicates that the variable can be calculated. Select it to perform the calculation.

E. Buttons: while there may be more than two buttons, these two exist in every template:

- Done or OK: exits the template.
- Clear or C: clears the contents of the template. This reverts the data to its defaults.

4.4.3 Types of Variable Data

There are 5 variable data types currently available. Variable data appears in the middle column of the template and can be selected and edited if it is underlined (otherwise it can only be calculated). Each variable data type is discussed here:

Numbers

Numbers are entered using the pop-up calculator. To enter a number:

- select the variable's data.
- enter the number with the keypad or use the calculator to compute it.
- select the "✓" (save button) to return to the template and save the number or choose "x" (cancel button) to return to the template without saving.

See the Using the Calculator : Interface Overview section for more information on the pop-up calculator.

Lists

Lists show a set of defined options. To choose an item from a list:

- select the variable.
- choose an option from the list, scrolling if necessary.

Tables

To edit table data:

- select the variable.
- enter data into the table by selecting each cell and entering data with the same pop-up calculator used with Numbers described above.
- select "OK" to save the data or "Cancel" to throw out those changes.

Dates

Dates are entered using a selector similar to other applications. On Palm OS handhelds, however, the year can be changed directly and in the range 1900 to 3000. To choose a date:

- select the variable.
- choose a year.
- choose a month.
- choose a day.

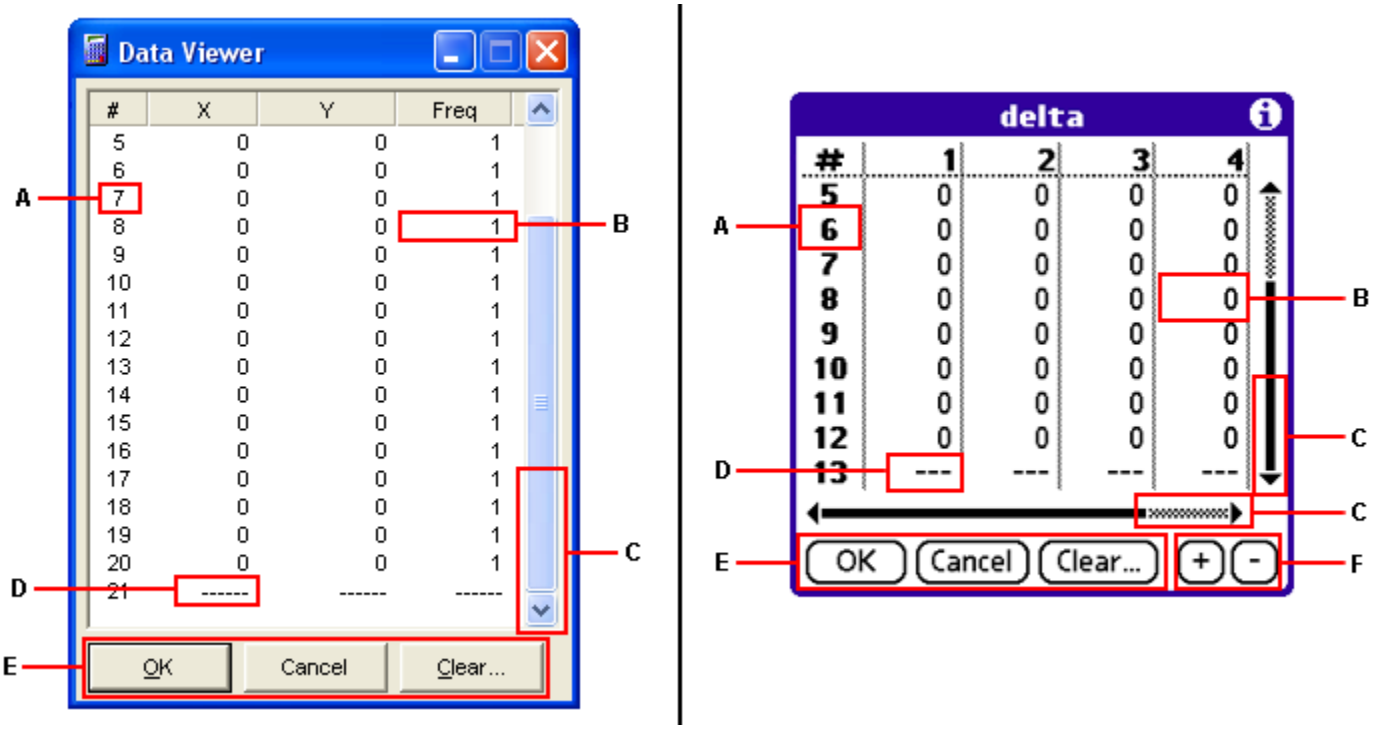
Times

Times are entered using a selector similar to other applications. To choose a time:

- select the variable.
- choose the hour by highlighting it and using the up/down scroll arrows.
- choose minutes by selecting each box in turn and using the up/down scroll arrows.
- choose am or pm (if not 24 hour time).
- select "OK" to save the data or "Cancel" to throw out those changes.

4.4.3.1 Tables

Table Editor



A. Row Header: the number of the row. Select a row header for a list of options:

- Insert: insert a row before the selected one.
- Delete: delete the selected row.
- Clear: clear the contents of the selected row, but does not delete it.

B. Cell: select a cell to edit its contents. Each cell is referenced by row-column coordinates. Use the pop-up calculator to enter data into the table, select checkmark to save or x to cancel. See the Templates : Using Templates : Data Entry section for more information on the pop-up calculator.

C. Scroll Bars: to see additional rows or columns, move the appropriate scroll bar. The scroll bar(s) only appear if there is more than one page of columns or rows. See Expand/Contract Buttons (G) to show more or less data on the screen at one time.

D. New Row: select "---" to enter a new data point at that row. This will create a new row, pushing "---" to the next row.

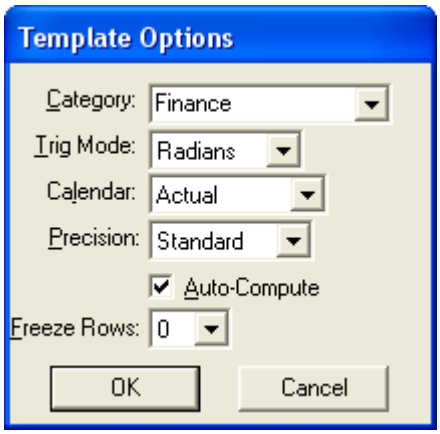
E. Buttons: select "OK" to save changes and "Cancel" to delete changes and leave the table editor. Select "Clear" to clear the contents of the entire table.

F. Expand/Contract Buttons: select the "+" button to see more columns on the screen at one time and "–" button to see fewer columns on the screen at one time (Palm OS only). You can scroll left and right and expand/contract columns by hand on Windows and PPC.

4.4.4 Template Preferences

These preferences impact the entire template.

Category
In My Templates and Template List, templates can be sorted into categories. Templates can be moved to any category. Choose a category from the list or select "New Category" to create a new one. Categories only exist as long as there are templates in them. To delete a category, move all the templates in that category to another category or delete all templates in that category.



Trig Mode
Calculates trigonometric functions as either radians or degrees.

Calendar
Date arithmetic can be performed using an actual year, 30 day month/360 day year, actual/360 or actual/365 calendar. Some templates allow for this change directly in the template itself.

- Actual: counts the actual number of days in a month and the total number of days in a year, including leap years.
- 30/360: treats all months as though they have 30 days and years as though they have 360 days.
- Actual/360: counts the actual number of days in a month, but treats years as though they have 360 days.
- Actual/365: counts the actual number of days in a month, but treats years as though they have 365 days.

These different day count methods are sometimes referred to as "basis" or "day count basis" and are frequently used in financial calculations.

Precision
Some calculations require an iterative search for the answer (e.g., TVM interest rates) and take more than 1 second to calculate. When this occurs, a Computing dialog will appear. Precision affects the time of the calculation. Setting it to "Full" will calculate until full precision is reached. Setting it to "Standard" will only calculate to the displayed decimal places. Full precision takes longer to calculate but is more accurate than standard precision.

Auto-Compute
When auto-compute is checked, the application will automatically determine when the "?" should be visible. When a value is entered into the template, a "?" will appear next to any other variable that may be affected by the change. When a value is recomputed, the question marks will disappear. While auto-compute is checked, any other values that can automatically be recalculated will be after each change.

For example, if a template contains just two variables (such as most conversion templates), changing the value of one variable will automatically cause the other variable to be recalculated. If unchecked, the compute "?" will remain visible at all times, and values will only be recalculated when the "?" button is selected.

Freeze Rows
When a template is long, often it is handy to freeze a certain number of rows at the top. These rows will always remain visible. 0 means no rows remain frozen, 1 means the first row at the top of the template will always remain visible, and so on.

4.4.5 Variable Preferences

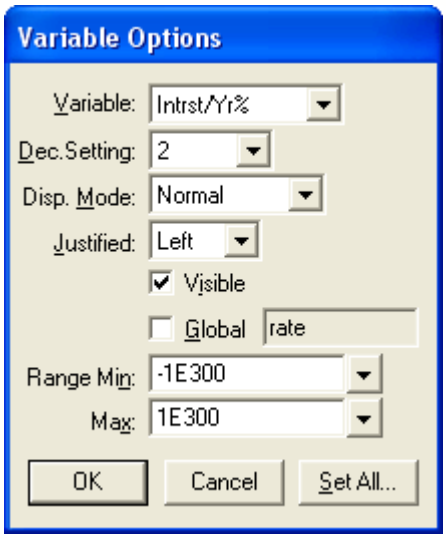
These preferences impact each variable independently. To set the preferences for a variable, first choose it from the Variable pop-up list. Selecting "Set All" will set all the variables to the current variable's preferences.

Variable
The variable to set. The preferences change with different selected variables.

Dec Setting
The number of decimal places to display.

Float shows all available decimal places. 0 through 11 shows that many decimal places. With very large numbers, fewer decimal places may be displayed because of the total number of places available to show in the view window.

This is hidden for lists, tables, dates and times.



Disp Mode
Display numbers in normal, scientific or engineering notation.

Normal mode displays numbers as would normally be written on paper or, if the number is too large or too small to display all places, in scientific notation. Scientific mode displays numbers in 3.45e67 format. The number of places displayed after the decimal point is determined by the decimal setting. Engineering mode uses the decimal setting to determine a number of displayed significant digits and then adjusts the exponent to be a multiple of 3. The number of significant digits is 1 plus the decimal setting.

This is hidden for lists, tables, dates and times.

Justified
Show the variable's data left or right justified within the template.

Visible
If checked, the variable and its data will be visible in the template; otherwise, it will be hidden from view. A variable that is hidden will not be recalculated even if "Auto-Compute" is checked. It will also not be cleared if the "Clear" button is selected.

Global
If checked, global allows the variable's data to be shared with other templates. The equation variable, which may be different than the variable's label, is listed next to the "Global" checkbox. For this variable to be shared with another template, the other template must contain the equation variable exactly as shown in the first template.

Range
Range is available if the variable requires an iterative search to calculate its value. The range is the maximum and minimum starting points for calculation. The closer these are, the faster and more accurately a value can be derived.

This is hidden when the variable does not need to be solved with an iterative process. See Creating Templates : How the Solver Works for more information.

4.4.6 Sharing Templates & Data

This section discusses sharing templates and template data.

Export/Beam
Palm OS: export/beam templates or its data.

- Select the template to share.
- Choose "Export/Beam" from the list if the template can be shared. If it cannot be shared, this option does not appear. Export/Beam Options appears.
- Select the desired export/beam option.
- Follow the on-screen directions, if any are required.

Pocket PC/Windows: export templates or its data.

- Select the template to export.

- Choose the "Template" menu.
- Select the desired export option. If it cannot be shared, this option is not available.
- Follow the on-screen directions, if any are required.

To beam on Pocket PC, save the results to a text file, navigate to that file and use your Pocket PC's capabilities to beam the file. Additional information may be available in your Pocket PC user's guide.

The data export/beam options that come with the software:

- **Export template to file:** save the selected template in a file that can be synchronized to the desktop for archival or sharing purposes.
- **Export category to file:** save the current category and all associated templates in a file that can be synchronized to the desktop for archival or sharing purposes.
- **Beam template:** beam the selected template to another handheld that has this software.
- **Beam category:** beam the current category and all associated templates to another handheld that has this software.
- **Results to Memo Pad/Text File**: saves the resulting calculations to the Memo Pad or a text file.
- **Print Results**: sends the resulting calculations to a printer (Windows only).
- **Copy Results**: saves the resulting calculations to the system clipboard.

Infinity Softworks may offer additional export/beam plug-ins from its web site (Palm OS only). These plug-ins could include ones to communicate with word processors, spreadsheets and probe systems, among others. See the Plug-ins web page at www.infinitysw.com/finance for more information.

Import
Import templates.

- Select "Import" at the bottom of My Templates.
- Select the desired import option.
- Follow the on-screen directions, if any are required.

Templates, generally, are imported automatically when the software is first started. On Palm OS handheld computers, any template uploaded to the device will be imported. On Pocket PC and Windows, any template installed to the "powerOne" directory in "My Documents" will be installed automatically. If the template is on an expansion card, CD-ROM or not in the powerOne directory on Pocket PC and Windows computers, use the Import option in My Templates to install.

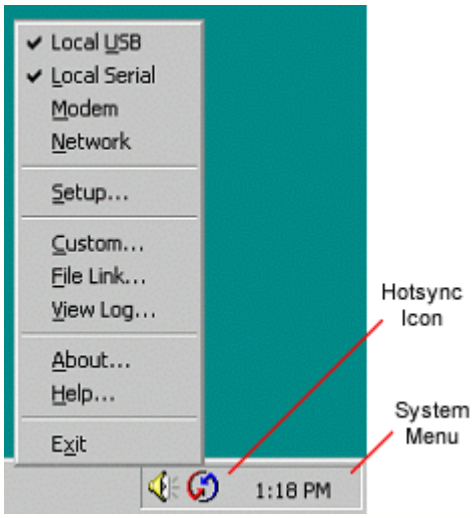
4.4.7 Synchronizing Templates

If you own both handheld and Windows versions of the software, it is possible to synchronize templates and data.

Palm OS

By default, synchronization happens each time the HotSync button is selected on the cradle or on the device.

For Palm OS devices, synchronization can be turned off, set to copy to the handheld only, to the Windows computer only, or synchronized both directions (two-way sync). With two way synchronization, the more recently changed template will be transferred. To change this preference for the next HotSync operation, tap the HotSync icon in the system tray of your desktop computer (which is usually in the bottom right corner of the screen):



Select "Custom…" from within the "HotSync" menu in your Palm Desktop software. Double-click the powerOne application and change your preference:

Pocket PC

The templates are synchronized each time the sync icon is selected in the desktop version of the software. See the Using the Calculator : Interface Overview : Display section for more information.

4.5 Included Templates

These templates are included with the software. Additional templates are available to download and install from Infinity Softworks' web site: www.infinitysw.com/finance.

Included template are sorted into the following categories:

Business
- Breakeven
- Discount
- Markup
- Percent Change
- Percent Total
- Profit Margin
- Sales Tax
- Summation
- Tip

Calendar
- Date
- Time

Convert
- Area
- Currency
- Length
- Mass
- Temperature
- Volume

Finance
- Bond
- Cash Flow
- Depreciation

- Interest Conv.
- Simple Interest
- TVM

Stats

- 1-Var Stats
- 2-Var Stats

4.5.1 One (1) & Two (2)

This section covers included templates beginning with the number 1.

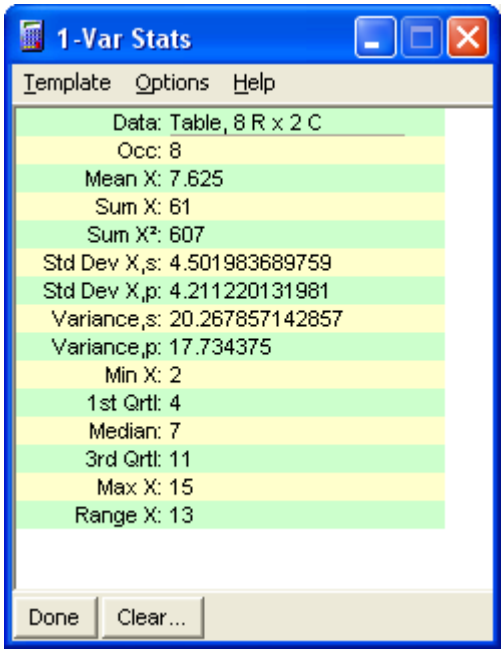
4.5.1.1 1-Variable Statistics

This template is for 1-variable descriptive statistics. It is in the Stats category. For information on accessing this template, see the Templates : Accessing section. For more on using templates in general, see the Templates : Using the Templates section.

Description
This template calculates descriptive statistics for one-variable data sets.

Variables

- **Data:** data set to analyze.
- **Occ:** total number of observations in the data set.
- **Mean X:** mean of x values. (Also known as weighted average.)
- **Sum X:** sum of x values.
- **Sum X^2:** sum of squared x values.
- **Std Dev X,s:** sample standard deviation (commonly denoted s).
- **Std Dev X,p:** population standard deviation (commonly denoted σ).
- **Variance,s:** sample variance (commonly denoted s^2).
- **Variance,p:** population variance (commonly denoted σ^2).
- **Min X:** minimum x value.
- **1st Qrtl:** median point between minimum and the median values, the 25th percentile. The calculation for finding the first quartile uses Tukey's method, which includes the median. This may differ from other calculators.
- **Median:** a middle value in the ordered data, the 50th percentile.
- **3rd Qrtl:** median point between the median and maximum values, the 75th percentile. The calculation for finding the third quartile uses Tukey's method, which includes the median. This may differ from other calculators.
- **Max X:** maximum x value.
- **Range X:** difference between minimum and maximum values.



This template automatically calculates a weighted average when the occurrences are different for each x data point.

Example

Given the following data:

Variable	Enter	Comments
Tap the Clear… button		Sets the display to its default values
Data Set	{5; 2; 8; 6; 3; 10; 12; 15}	
Frequency	1 Of Each	

Select "?" in the Occ row to calculate (may calculate automatically). The following answers are returned:

Variable	Calculated
Occ	8
Mean X	7.625
Sum X	61
Sum X^2	607
Std Dev X,s	4.5020
Std Dev X,p	4.2112
Variance,s	20.2679
Variance,p	17.7344
Min X	2
1st Qrtl	4
Median	7
3rd Qrtl	11
Max X	15
Range X	13

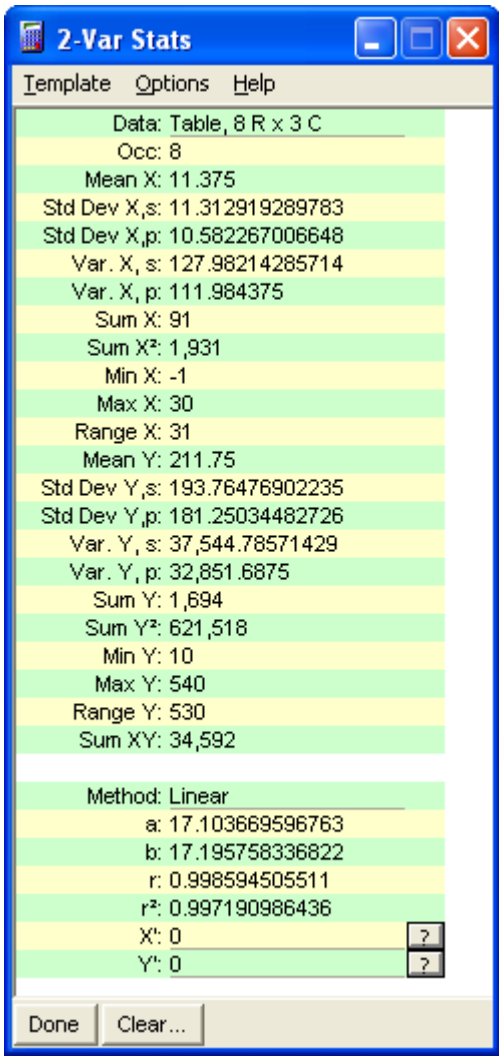
4.5.1.2 2-Variable Statistics

This template is for 2-variable descriptive statistics. It is in the Stats category. For information on accessing this template, see the Templates : Accessing section. For more on using templates in general, see the Templates : Using the Templates section.

Description
This template calculates descriptive statistics for two-variable data sets (variables x and y) with equal numbers of observations.

Variables

- **Data:** x and y data points. If desired, frequency of each (x, y) data point is also available.
- **Occ:** common number of observations in the two data sets (x data with frequency and y data with frequency must have the same occurrences).
- **Mean X:** mean of x values.
- **Std Dev X,s:** sample standard deviation of x values (commonly denoted s).
- **Std Dev X,p:** population standard deviation of x values (commonly denoted σ).
- **Var. X,s:** sample variance of x values (commonly denoted s^2).
- **Var. X,p:** population variance of x values (commonly denoted σ^2).
- **Sum X:** sum of x values.
- **Sum X^2:** sum of squared x values.
- **Min X:** minimum x value.
- **Max X:** maximum x value.
- **Range X:** difference between x minimum and maximum values.
- **Mean Y:** mean of y values.
- **Std Dev Y,s:** sample standard deviation of y values (commonly denoted s).
- **Std Dev Y,p:** population standard deviation of y values (commonly denoted σ).
- **Var. Y,s:** sample variance of y values (commonly denoted s^2).
- **Var. Y,p:** population variance of y values (commonly denoted σ^2).
- **Sum Y:** sum of y values.
- **Sum Y^2:** sum of squared y values.
- **Min Y:** minimum y value.
- **Max Y:** maximum y value.
- **Range Y:** difference between minimum and maximum y values.
- **Sum XY:** sum of x times y values.
- **Method:** regression method. See details below.
- **a:** regression y-intercept.
- **b**: regression slope.
- **r:** simple correlation coefficient.
- **r^2:** coefficient of determination.
- **X':** predicted x-value.
- **Y':** predicted y-value.



Regression Models

In general, it is always best to have as much data as possible when curve fitting.

- **Linear:** y = ax + b
- **Log (natural):** y = a + b * ln(x) for x > 0
- **Log (base 10):** y = a + b * log(x) for x > 0
- **Exponential:** y = ab^x
- **Power:** y = ax^b

Example

Given the following data:

Variable	Enter	Comments
Tap the Clear… button		Sets the display to its default values
Data X	{-1; 0; 7; 12; 4; 14; 25; 30}	
Data Y	{10; 32; 128; 213; 75; 250; 446; 540}	
Frequency	1 Of Each	

Select "?" in the Occ row to calculate (may calculate automatically). The following answers are returned:

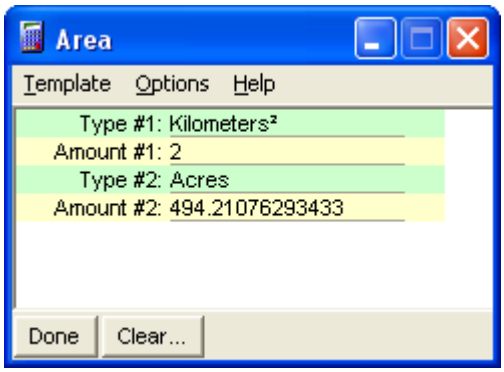
Variable	Calculated
Occ	8
Mean X	11.375
Std Dev X,s	11.3129
Std Dev X,p	10.5823
Var. X,s	127.9821
Var X,p	111.9844
Sum X	91
Sum X^2	1,931
Min X	-1
Max X	30
Range X	31
Mean Y	211.75
Std Dev Y,s	193.7648
Std Dev Y,p	181.2503
Var. Y,s	37,544.7857
Var. Y,p	32,851.6875
Sum Y	1,694
Sum Y^2	621,518
Min Y	10
Max Y	540
Range Y	530
Sum XY	34,592

4.5.2 A-D

This section covers included templates beginning with the letters A through D.

4.5.2.1 Area

This template is for Area conversions. It is in the Convert category. For information on accessing this template, see the Templates : Accessing section. For more on using templates in general, see the Templates : Using the Templates section.



Variables

- **Type #1:** the unit type to convert from.
- **Amount #1:** the amount of the first type.
- **Type #2:** the unit type to convert to.
- **Amount #2:** the amount of the second type.

Example

If the map states that the land's area is 2 km^2, what is its area in acres?

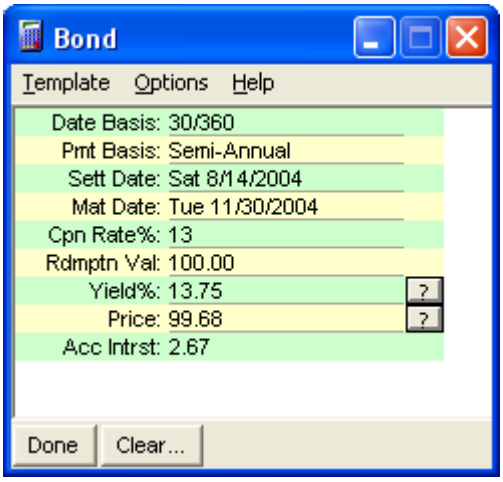
Variable	Enter	Comments
Tap the Clear… button		Sets the display to its default values
Type #1	Kilometers^2	Choose from the list
Amount #1	2	
Type #2	Acres	Choose from the list

Select "?" in the Amount #2 row to calculate (may calculate automatically). The area is 494.2 acres.

4.5.2.2 Bonds

Bond computations are used to calculate corporate or municipal bond investments. These computations include two day-count methods (actual or 30/360) and four coupon per year settings (once, twice, four, or twelve times per year).

Variables



- **Date Basis:** computations based on actual/actual month/year method or the 30/360 method.
- **Pmt Basis:** How often the coupon payments occur: 1, 2, 4, or 12 times per year.
- **Sett Date:** the settlement or purchase date.
- **Mat Date:** the maturity or call date. This date always occurs after the settlement date. This date is called a call date when the issuer can pay off the bond before the maturity date. Maturity date can also be called the redemption date.
- **Cpn Rate%:** the annual coupon rate as a percentage. This is the annual interest rate printed on the bond and is used to determine the coupon payment (the periodic payment of interest). This value is entered as a percentage. For example, 7.25% is entered as "7.25".
- **Rdmptn Val:** the redemption value is a percentage of the bond's par value that is paid to the owner when it is retired. If the calculation is to the maturity date, this value is 100. This is the standard set by HP. If the calculation is to a call date, this value varies. The par value is the value printed on the bond itself. A bond is often said to sell at a premium or discount. This is reflected in the redemption value. A bond that sells at a discount sells at less than par value. Bonds that sell at a premium are for more than par value.
- **Yield%:** the yield to maturity or redemption. This is the rate of return to the investor based on earnings from payments of principal and interest. This includes a sale at a premium or discount. To calculate yield, a value for price must be entered. This value is entered as a percentage. For example, 8.385% is entered as "8.385".
- **Price:** the dollar price. To calculate the dollar price, a value for yield must be entered.
- **Acc Interest:** the accrued interest based on $100 of par value. This value is calculated automatically when computing either yield or price.

Example

A corporate bond matures on November 30, 2004 with a settlement date of August 14, 2004. It pays 13% coupon on a semi-annual basis, with a 30/360 day-count method. It will be redeemed at 100% of par and an annual yield of 13.75%. What is the price and accrued interest?

Variable	Enter	Comments
Clear…		Sets the display to its default values
Date Basis	30/360	
Pmt Basis	Semi-Annual	
Sett Date	8/14/04	August 14, 2004
Mat Date	11/30/04	November 30, 2004
Cpn Rate%	13	
Rdmptn Val	100	
Yield%	13.75	

Compute the price by selecting "?" on the same line. The accrued interest computes automatically. The price is $99.68 and the accrued interest is $2.67.

4.5.2.3 Breakeven

For any company, making a profit is the key to success. By analyzing the relationship between revenues and expenses, the levels at which a company has to operate in order to break even can be determined. Breakeven is the point at which expenses equal revenues. Until that point, a company is operating at a loss.

Variables

- **Fixed Cost:** the fixed costs. These are costs that are not dependent on each unit sold. An example is rent – whether 0 or 5000 units are sold, the rent will always be the same.
- **Var Cost:** the variable cost per unit. These are costs that are dependent on each unit sold. For instance, shipping costs do not occur unless a unit is sold.
- **Price:** the price per unit. This is the price at which the product is sold.
- **Tax Rate%:** applicable tax rate as a percentage.
- **Profit:** the amount of profit determined or expected. Positive values are profits while negative ones are losses.
- **Quantity:** the number of units sold.

Example

A startup company has $500,000 in operating expenses every month. It is introducing its first product, which costs $115 to produce. This product will sell to distributors for $245 per unit. Its tax rate is 40%. How many units must the company sell every month to cover its costs (break even)?

Variable	Entry	Comments
Clear…		Sets the display to its default values
Fixed Cost	500,000	
Var Cost	115	
Price	245	
Tax Rate%	40	
Profit	0	

Compute the quantity by selecting "?" on the same line. The company needs to sell 3,847 units per month.

4.5.2.4 Cash Flows

This template is for Cash Flow calculations. See the Included Templates section for category information.

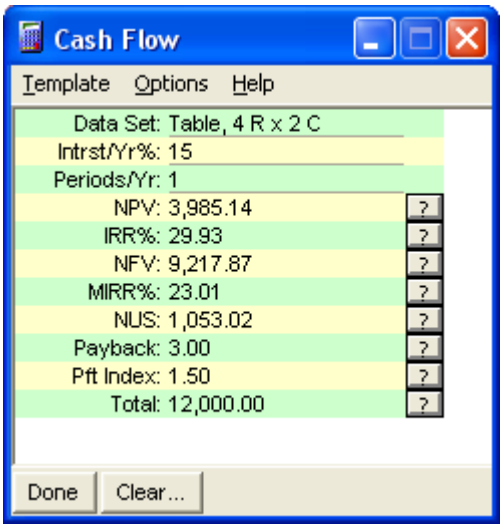
Description

The Cash Flow template analyzes financial investments involving outflows and inflows of cash which occur on a regular basis but do not necessarily occur in similar amounts.

As with other templates, the Cash Flow template understands positive numbers to be inflows of cash (cash received) and negative numbers to be outflows (cash paid). Note that, although the interval between cash flows must be equal, the amounts of those cash flows do not have to be the same. Cash flows generally involve some initial outflow of cash followed by subsequent inflows over a number of periods. For instance, an initial outflow (designated by location 0) could be followed by various amounts paid back over five periods, each period being one year apart. Cash flows do not have to be initial outflows followed by inflows, either. Cash flows could be inflows preceded by outflows, or an initial inflow or outflow followed by some mixture of cash flows as well. See the Understanding Cash Flows section for a pictorial explanation of inflows and outflows.

Variables

- **Amounts:** Data set of cash flow values. The 0^{tn} cash flow is the initial cash flow.
- **Frequency**: Corresponding set of amount occurrences. This list must be the same length as Amounts. The template assumes a frequency of 1 per amount entry.
- **Intrst/Yr%:** Required to perform some cash flow computations, the interest per year is entered as a percentage. For instance, 8.25% would be entered as "8.25".
- **Periods/Yr:** Periods per year automatically adjusts the interest rate. When the interest rate is entered as a yearly rate, entering a value for periods per year will automatically determine the periodic interest rate. When calculating an interest rate (with IRR or MIRR), the reported interest rate will be on a yearly basis (the number of periods per year entered times the rate).



Calculations

- **NPV:** The net present value (NPV) method computes the amount gained or lost on a given investment in today's dollars. This uses a market rate of return (interest per year) to discount cash flows back to the present. Assuming an initial cash outflow, a positive NPV means the investor's assets would increase and the investment should be attractive. A negative NPV means that the investor's assets would decrease and the investment is not attractive. If NPV is zero, then the investor would probably be neutral to the investment. If the initial cash flow is an inflow, the reverse would be true.
- **IRR%:** The internal rate of return (IRR%) computes the rate at which the investment pays for itself. This can be compared against a desired rate of return. If the IRR is greater than a desired rate, the investment may be attractive. The internal rate of return method does not take interest per year or periods per year into consideration when calculating.
  The internal rate of return calculation is very complicated. Calculating IRR uses an iterative approach to solving the problem and, if there is an answer, may take quite some time to calculate. A few caveats to calculating internal rate of return exist. First, long calculations may be interrupted because an iterative limit is exceeded within the calculator itself. Errors may occur in other areas as well. If there is no sign change within the cash flow problem, an error will occur. If the cash flow will yield a negative IRR amount, an error will occur. A negative IRR means that there is at least one negative answer and possibly multiple negative and positive answers to the same cash flow question. In these situations, the calculator will not display an answer. Finally, if there are multiple sign changes (two or more) within the same problem, there may be multiple solutions. Note that multiple sign changes may not give an answer for IRR. Also, if IRR is less than zero it does not give an answer on IRR. The calculator gives the IRR closest to 0, but extreme caution should be used in basing an investment on this type of cash flow. See the Understanding Cash

Flows section for more information.
Another method for solving IRR problems is by estimating an interest per year value and calculating net present value (NPV). Internal rate of return is calculated by solving for NPV when it is equal to 0. With this in mind, by estimating an interest per year amount, you can solve for the internal rate of return. The closer to a net present value of zero, the more accurate the IRR estimate becomes.

- **NFV:** The net future value (NFV) computes the future value of the net present value.
- **MIRR%:** The modified internal rate of return (MIRR%) is an alternative for IRR when there is more than one sign change. When IRR has multiple sign changes, IRR can have more than one answer. MIRR eliminates sign changes by using reinvestment and borrowing interest rates instead (the Cash Flow template assumes these rates are the same).
- **NUS:** The net uniform series (NUS) performs computations by taking the net present value of the cash flows if they are even and regular.
- **Payback:** The payback method tells at which period an initial investment will be paid back. If there is no payback, the reported answer is 0. The payback method does not take interest per year or periods per year into consideration when calculating.
- **Pft Index:** The profitability index, also known as the benefit/cost ratio, shows the relative profitability of any cash flow problem, dividing the present value of the inflows by the present value of the outflows.
- **Total:** The total is the sum of the cash flows.

Examples

Your company is looking to buy a new piece of equipment to help it increase manufacturing capacity to meet demands for its largest product. The managers are wondering what the return would be if the equipment was purchased for $8,000. You can expect at least a 15% return on the investment elsewhere and are counting on the following yearly cash flows: Year #1: $2,000, Years #2-#3: $3,000 each year, Years #4-#6: $4,000 each year. What are each of the cash flow computations for comparison?

Variable	Selection/Entry	Comments
Clear…		Sets the display to its default values
Amounts		Select the label to create or select a table
Table	None Selected	Select None Selected and choose New...
Name	CFData	Name the table
Size, rows	1	
Size, columns	2	One column for Amounts, one for frequency
OK		

Enter the cash flow items in the Table Editor

#	1	2	
0	–8,000	1	Initial cash outflow, which occurs once
1	2,000	1	Next cash flow item is $2,000 inflow which occurs once in year 1
2	3,000	2	Next item is $3,000 inflow which occurs twice, one time each in years 2 and 3
3	4,000	3	Next item is $4,000 inflow which occurs three times, one time each in years 4, 5 and 6

Select "OK" to save the cash flow data and "OK" again to return to the cash flow template. Note that Amounts is set to CFData, Col 1 automatically. Continue entering data:

Variable	Selection/Entry	Comments
Table	None Selected	Select None Selected and choose CFData
Column	2	Select the second column for frequency
OK		
Intrst/Yr%	15	
Periods/Yr	1	

Calculating each of the remaining variables in the template yields the following answers:

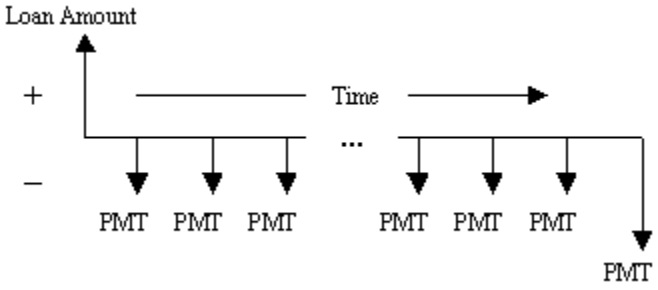
- NPV = $3,985.14
- IRR% = 29.93%
- NFV = $9,217.87
- MIRR% = 23.01%
- NUS = $1,053.02
- Payback = 3 periods (in this case years because periods per year is one)
- Pft Index = 1.50
- Total = $12,000

4.5.2.4.1 Understanding Cash Flows

To further understand the cash flow model, here is an example of a timeline. Note that inflows of cash are treated as positive amounts (designated by a [+] sign) and outflows of cash as negative amounts (designated by a [–] sign).

The Cash Flow and Time Value of Money templates both use cash flows. The difference is in the entry and interpretation. The Time Value of Money template deals with cash flows as annuities. Each of these cash flows are the same amount. The loan, lease and regular deposit examples on the next page are annuity problems solved in the TVM template. The Cash Flow template deals with investments where the payment is in varying amounts.

This example shows a typical loan problem, where the initial cash flow, the loan amount, is an inflow. Each of the subsequent cash flows - payments to the bank - are cash outflows.



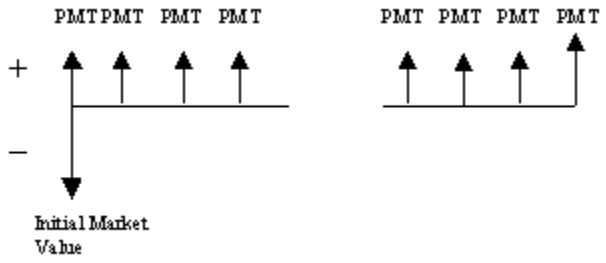
A few items to note:

- The length between cash flows is the same. This denotes that inflows and outflows occur at regular intervals of time.
- This cash flow begins with an inflow followed by subsequent outflows of cash. The cash flow can, however, begin with an outflow and be followed by subsequent inflows of cash. Furthermore, there can be mixed inflows and outflows of cash.
- The payment amounts are the same length, meaning that each payment is the same amount. This could differ for Cash Flow template problems as explained above because of the possibility for varying sized cash flows. Assume that, at the end of the series of cash flows, there was some larger payment (called a balloon payment) to pay off this loan because the last cash flow is longer than the others.

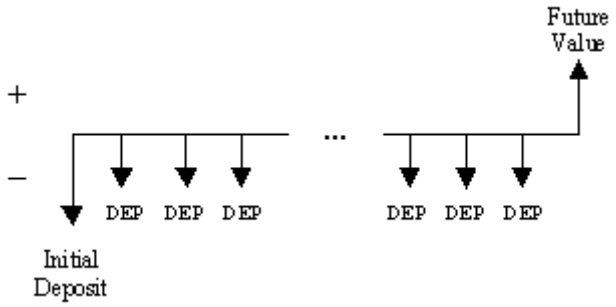
All cash flow problems can be represented in this fashion, with cash inflows and outflows viewed over some time period. The following are examples of other types of cash flow or TVM problems:

Examples

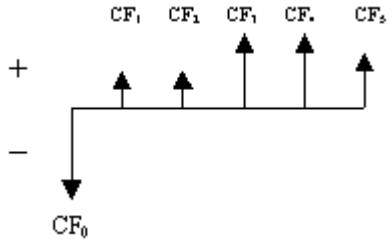
Leases



Investment with regular deposits

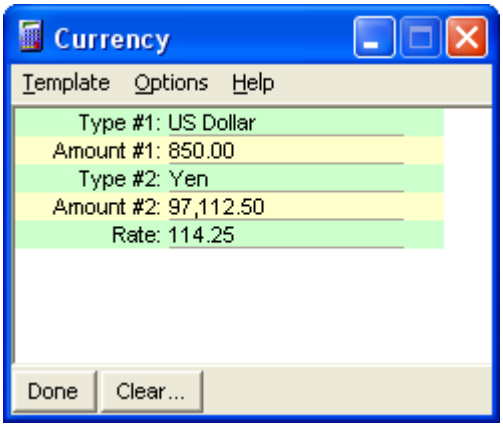


Cash flow with one sign change



4.5.2.5 Currency Conversions

This template is for Currency conversions.



Variables

- **Type #1:** the unit type to convert from.
- **Amount #1:** the amount of the first type.
- **Type #2:** the unit type to convert to.
- **Amount #2:** the amount of the second type, calculated.
- **Rate:** conversion rate. In general, Amount #1 x Rate = Amount #2. Each time a currency conversion is calculated, the conversion's rate is stored automatically and recalled when those two currencies are set for Type #1 and Type #2. To clear that stored value, recall the two currencies and select Clear.

Example

The exchange rate from US dollars to Japanese yen is 114.25. If you are exchanging $850, how many yen do you have?

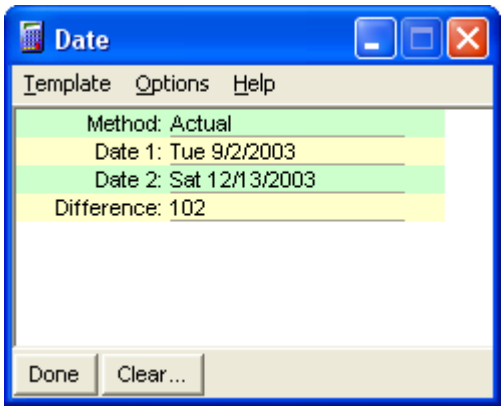
Variable	Enter	Comments
Tap the Clear… button		Sets the display to its default values
Type #1	US Dollars	Not used in computation
Amount #1	850	
Type #2	Yen	Not used in computation
Rate	114.25	

Compute the amount of yen by selecting "?" on the Amount #2 line. You would receive 97,122.50 yen.

4.5.2.6 Date

This template is for Date calculations. It is in the Calendar category. For information on accessing this template, see the Templates : Accessing section. For more on using templates in general, see the Templates : Using the Templates section.

Variables



- **Method:** the day-count method. Date arithmetic can be performed using an actual year, 30 day month/360 day year, actual/360 or actual/365 calendar.
  - Actual: counts the actual number of days in a month and the total number of days in a year, including leap years.
  - 30/360: treats all months as though they have 30 days and years as though they have 360 days.
  - Actual/360: counts the actual number of days in a month, but treats years as though they have 360 days.
  - Actual/365: counts the actual number of days in a month, but treats years as though they have 365 days.
- **Date 1:** the date to compute from.
- **Date 2:** to date to compute to.
- **Difference:** the difference in number of days.

Example

Vacation begins on December 13, 2003. Today is September 2, 2003. How many actual days until vacation?

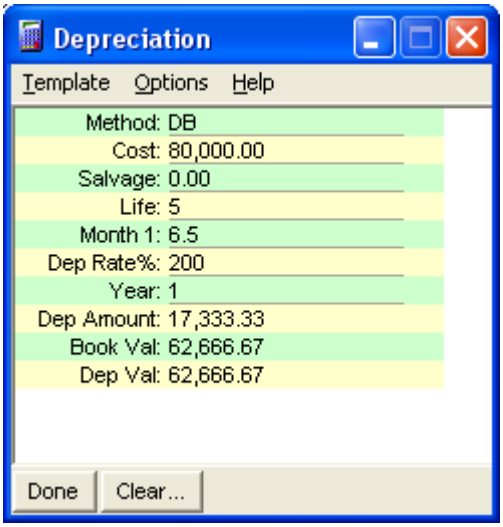
Variable	Enter	Comments
Tap the Clear… button		Sets the display to its default values
Method	Actual	Choose from the list
Date 1	9/2/03	Enter September 2, 2003
Date 2	12/13/03	Enter December 13, 2003

Select "?" in the Difference row to calculate. There are 102 days until vacation.

4.5.2.7 Depreciation

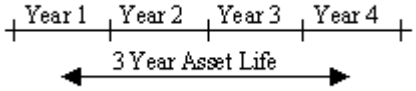
Depreciation is an important source of revenue reduction in businesses. The four most common depreciation methods for book purposes are available: straight-line, declining balance, sum-of-the-year's digits, and declining-balance crossover. A separate template provides ACRS depreciation, often used for tax purposes.

Variables



- **Method:** depreciation method. See Calculation Methods below for more information.
- **Cost:** the cost to purchase the asset.
- **Salvage:** the assumed value of the asset at the end of the asset's life.
- **Life:** the length of time the asset will be in service.
- **Month 1:** the first month the asset will be placed in service where January is 1 and December is 12. Entering "6.5", for example, means the asset was placed into service half way through the sixth month (approximately June 15).
- **Dep Rate%:** the declining balance rate. This is used in DB and DB x SL calculations. This is entered as a percentage. For example, 200% declining is entered as "200".
- **Year:** year to calculate depreciation.
- **Dep Amount:** amount of depreciation for the year.
- **Book Val:** original cost of the asset less accumulated depreciation. Accumulated depreciation is the total depreciation taken through the calculated year. This is the value of the asset remaining on the company's books.
- **Dep Val:** depreciation value. This is the book value less the salvage value for the asset.

Because an asset can begin depreciation on a date other than the first of the year, the calendar life may be greater than the amount entered for Asset Life. For instance, if an asset is expected to have a useful life of 3 years, beginning in March (the third month), the last calendar year is actually the fourth year:



Calculation Methods

The straight-line (SL) method depreciates the same amount every year of the asset's life.

The declining balance (DB) method depreciates more in the first few year's of the assets life than in the later years. This method, along with the DB x SL method, uses the declining balance rate to calculate the depreciation value.

The declining balance cross straight-line (DB x SL) method is often used for tax purposes. In this instance, the declining balance method is used until the optimal time to switch to the straight-line method. The calculator determines this point when depreciation is higher using the straight-line method than the declining balance method. This method also uses the declining balance rate for depreciating.

The sum of the year's digits (SOYD) method, like declining balance, allocates more depreciation to the early years of the asset's life. This method uses a complex formula based upon the number of years the asset will be in service to determine a depreciation rate.

Example

$80,000 worth of equipment was recently purchased in the middle of June. With a five-year useful life and no salvage value, these computers will be depreciated using the declining balance method at a 200% rate. What is the depreciation amount for the first year?

Variable	Enter	Comments
Clear…		Sets the display to its default values
Method	DB	
Cost	80,000	
Salvage	0	
Life	5	
Month 1	6.5	Half way through the sixth month
Dep Rate%	200	

Compute the depreciation information by entering the year and selecting "?". The depreciation amount will be $17,333.33.

4.5.2.8 Discount

This template is for Discount calculations. It is in the Business category. For information on accessing this template, see the Templates : Accessing section. For more on using templates in general, see the Templates : Using the Templates section.

Variables

- **Price:** the price before discounts.
- **Sales Price:** the sales price after discount.
- **Discount%:** percentage discount. For example, a 30% discount would be entered as "30".
- **Difference:** the difference between the price and the sales price.

Example

The coupon is for 10% off the original cost. What is the discounted price if the cost is $9.99?

Variable	Enter	Comments
Tap the Clear… button		Sets the display to its default values
Price	9.99	
Discount%	10.00	
Difference	1.00	

Select "?" in the Price row to calculate. The sales price is $8.99.

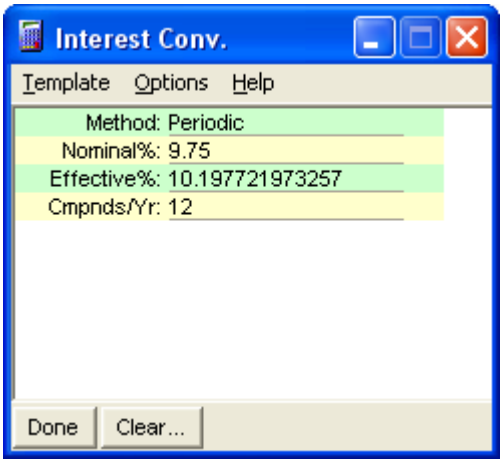
4.5.3 E-P

This section covers included templates beginning with the letters E through M.

4.5.3.1 Interest Conversions

Comparing interest rates may be necessary when two investment possibilities present themselves. Investments are usually stated in terms of an annual, nominal interest rate (or annual percentage rate) but each investment often has a different number of compounding periods per year. To compare these investments, the interest rates must first be converted to an annual, effective interest rate.

Variables



- **Method:** conversion method: either continuous or periodic. With periodic interest conversion, there is a set number of compounding periods per year, such as quarterly (4 times per year), monthly (12), or yearly (1). With continuous compounding, there is no set number of periods per year.
- **Nominal%:** the annual, nominal interest rate expressed as a percentage. For example, 8.25% is entered as "8.25".
- **Effective%:** the annual, effective interest rate expressed as a percentage. For example, 8.25% is entered as "8.25".
- **Cmpnds/Yr:** the number of compounding periods per year. For example, if interest is compounded quarterly, this value would be set to "4". Compounding periods per year is used only when the method is set to periodic.

Example

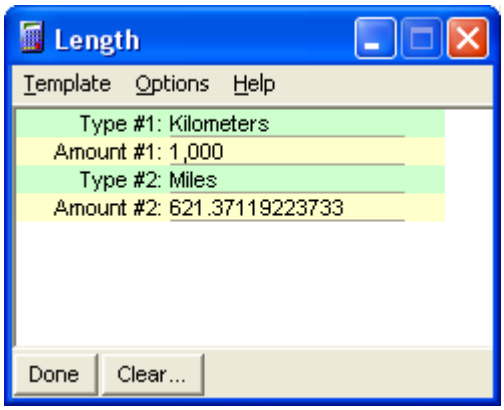
You are presented with two competing investments. The first is compounded monthly with a nominal interest rate of 9.75%. The other pays at an effective interest rate of 10%. Which investment has a better interest rate?

Variable	Enter	Comments
Clear…		Sets the display to its default values
Method	Periodic	
Nominal%	9.75	
Cmpnds/Yr	12	

Compute the effective rate by selecting "?" on the same line. The first investment's effective rate is 10.20%. It has the better interest rate.

4.5.3.2 Length

This template is for Length conversions. It is in the Convert category. For information on accessing this template, see the Templates : Accessing section. For more on using templates in general, see the Templates : Using the Templates section.



Variables

- **Type #1:** the unit type to convert from.
- **Amount #1:** the amount of the first type.
- **Type #2:** the unit type to convert to.
- **Amount #2:** the amount of the second type.

Example

It is 1,000 kilometers from your home to your aunt's house. How many miles is this?

Variable	Enter	Comments
Tap the Clear… button		Sets the display to its default values
Type #1	Kilometers	Choose from the list
Amount #1	1000	
Type #2	Miles	Choose from the list

Select "?" in the Amount #2 row to calculate (may calculate automatically). The conversion is 621.4 miles.

4.5.3.3 Markup

This template is for Markup calculations. It is in the Business category. For information on accessing this template, see the Templates : Accessing section. For more on using templates in general, see the Templates : Using the Templates section.

Variables

- **Method:** computation based on price or cost. Profit margin computations are based on price; percent change computations are based on cost.
- **Cost:** the cost to manufacture or purchase.
- Price: the selling or resale price.
- **Markup%:** the markup expressed as a percentage. For example, an 8.125% change would be entered as "8.125". A positive value represents an increase while a negative one represents a decrease.
- **Difference:** the difference between the price and the cost.

Example

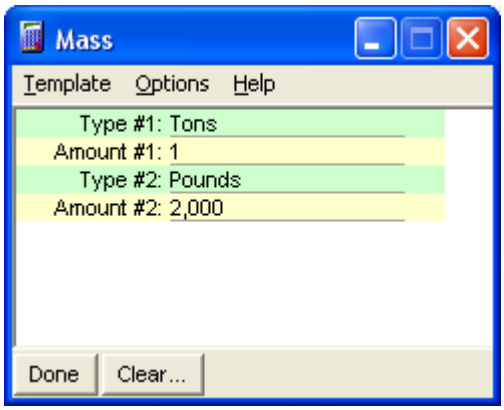
The clothing is sold at a 25% markup on cost. What is the price if the cost is $29.99?

Variable	Enter	Comments
Tap the Clear… button		Sets the display to its default values
Method	% of Cost	Choose from the list
Cost	29.99	
Markup%	25.00	
Difference	7.50	

Select "?" in the Price row to calculate. The price is $37.49.

4.5.3.4 Mass

This template is for Mass conversions. It is in the Convert category. For information on accessing this template, see the Templates : Accessing section. For more on using templates in general, see the Templates : Using the Templates section.



Variables

- **Type #1:** the unit type to convert from.
- **Amount #1:** the amount of the first type.
- **Type #2:** the unit type to convert to.
- **Amount #2:** the amount of the second type.

Example

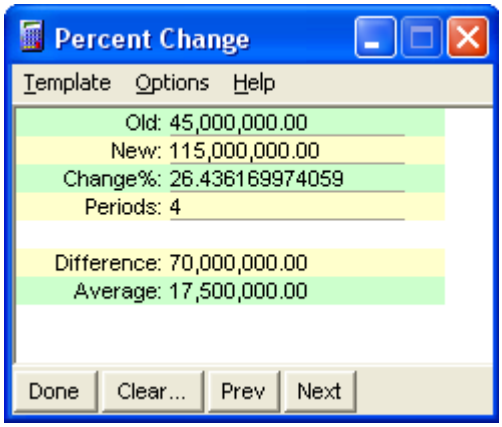
The elephant weighs 1 ton. How many pounds is this?

Variable	Enter	Comments
Tap the Clear… button		Sets the display to its default values
Type #1	Tons	Choose from the list
Amount #1	1	
Type #2	Pounds	Choose from the list

Select "?" in the Amount #2 row to calculate (may calculate automatically). The elephant weighs 2,000 pounds.

4.5.3.5 Percent Change

This template is for Percent Change calculations. It is in the Business category. For information on accessing this template, see the Templates : Accessing section. For more on using templates in general, see the Templates : Using the Templates section.



Variables

- **Old:** the old value.
- **New:** the new value.
- **Change%:** the percentage changed per period. For example, an 8.125% change would be entered as "8.125". A positive value represents an increase while a negative one represents a decrease.
- **Periods:** the number of periods.
- **Difference:** the difference between the old and new amounts.
- **Average:** the avarage of the old and new amounts.

Example

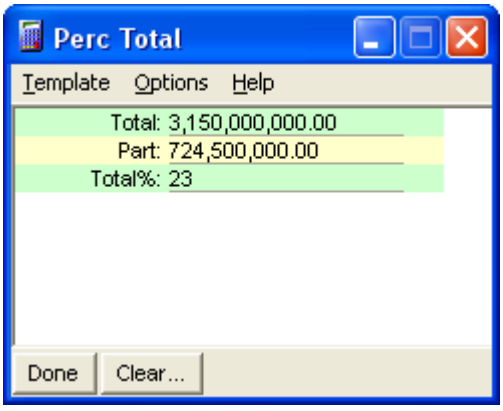
Over 4 years, sales increased from $45 million to $115 million. The industry average is 20% increase per year. How does your company compare?

Variable	Entry	Comments
Tap the Clear… button		Sets the display to its default values
Old	45,000,000	
New	115,000,000	
Periods	4	
Difference	70,000,000.00	
Average	17,500,000.00	

Select "?" in the Change% row to calculate. Sales have increased 26.44% per year, comparing favorably to the 20% industry pace.

4.5.3.6 Percent Total

This template performs percent total computations.



Variables

- **Total:** the total.
- **Part:** the portion of the total.
- **Total%:** the percentage of the total. For example, an 8.125% change would be entered as "8.125".

Example

Your division contributes 23% of the company's revenue. If total revenue is $3.15 billion, what is your division's contribution?

Variable	Entry	Comments
Clear…		Sets the display to its default values
Total	3,150,000,000	
Total%	23	

Compute the part by selecting "?" on the same line. Your division contributes $724,500,000 in revenue.

4.5.3.7 Profit Margin

This template performs profit margin computations.

Variables

- **Cost:** the cost to manufacture or purchase.
- **Price:** the selling or resale price.
- **Margin%:** the gross profit margin expressed as a percentage. For example, an 8.125% change would be entered as "8.125". A positive value represents an increase while a negative one represents a decrease.

Profit margin and markup as a percentage of price are identical.

Example

Your division contributes $724,500,000 in revenue. The costs associated with revenue are $580,000,000. What is the profit margin?

Variable	Entry	Comments
Clear…		Sets the display to its default values
Cost	580,000,000	
Price	724,500,000	

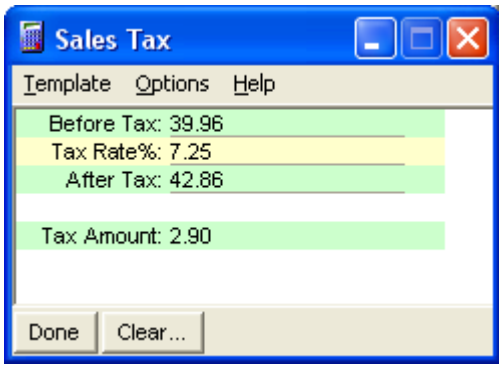
Compute the profit margin by selecting "?" on the Margin% line. The profit margin is 19.94%.

4.5.4 O-Z

This section covers included templates beginning with the letters N through S.

4.5.4.1 Sales Tax

This template is for Sales Tax calculations. It is in the Business category. For information on accessing this template, see the Templates : Accessing section. For more on using templates in general, see the Templates : Using the Templates section.



Variables

- **Before Tax:** before tax amount.
- **Tax Rate%:** tax rate expressed as a percentage. For example, a 6% tax rate would be entered as "6".
- **After Tax:** after tax amount.
- **Tax Amount:** the total amount of taxes.

Example

Dinner for two cost $39.96 before taxes. What is the bill after taxes if the tax rate is 7.25%?

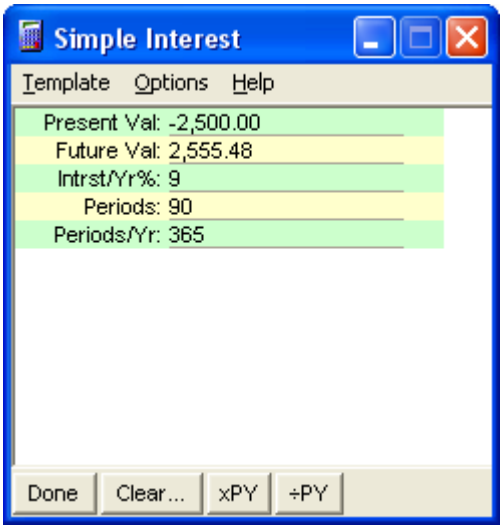
Variable	Entry	Comments
Tap the Clear… button		Sets the display to its default values
Before Tax	39.96	
Tax Rate%	7.25	
Tax Amount	2.90	

Select "?" in the After Tax row to calculate. The bill after taxes is $42.86.

4.5.4.2 Simple Interest

Simple interest problems assume that interest is accumulated only once (at the time of repayment). These computations are performed in the Simple Interest template. For compounding interest computations use the Time Value of Money (TVM) template.

Variables



- **Present Val:** the present or current value.
- **Future Val:** the future value.
- **Intrst/Yr%:** interest per year as a percentage. For example, 8.25% interest should be entered as "8.25".
- **Periods:** number of total periods. This number is the number of years and months times the periods per year. For example, if the loan is 4 years with 12 payments per year (monthly payments), periods should be 48 (4 x 12).
- **Periods/Yr:** the number of payment periods per year.

Buttons

- **xPY:** quick set button for the number of periods. This button multiplies the value in periods by the value in periods per year. For example, to convert 10 years at 12 periods per year to periods, enter 10 in periods and 12 in periods per year then select xPY.
- **÷PY:** quick set button for the number of periods. This button divides the value in periods by the value in periods per year. For example, if periods is 60 with periods per year equal to 12, discovering that is five years can be done easily by selecting ÷PY.

Example

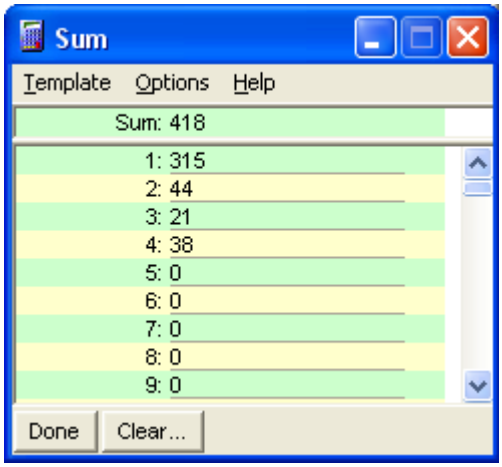
A good friend has asked for a 90-day loan of $2,500 to get involved with a real estate investment. You have agreed to lend him the money at 9% interest, calculated on a 365-day basis. What amount will be paid back at the end of this period?

Variable	Enter	Comments
Clear…		Sets the display to its default values
Present Val	–2,500	
Intrst/Yr%	9	
Periods	90	
Periods/Yr	365	

Compute the future value by selecting "?" on the same line. The friend should repay $2,555.48 in 90 days.

4.5.4.3 Summation

This template performs summation calculations.



Variables

- **Sum:** summation of the list of numbers.
- **1-100:** list of numbers. To subtract, enter it as a negative number.

Example

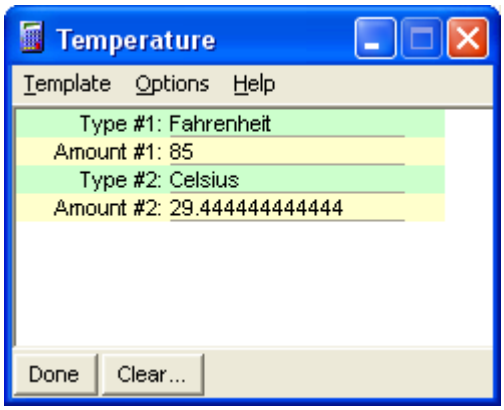
You have four expenses: $315, $44, $21, and $38. What is the total amount for the expense report?

Variable	Entry	Comments
Clear…		Sets the display to its default values
1	315	
2	44	
3	21	
4	38	

The Sum calculates automatically. It equals $418.

4.5.4.4 Temperature

This template is for Temperature conversions. It is in the Convert category. For information on accessing this template, see the Templates : Accessing section. For more on using templates in general, see the Templates : Using the Templates section.



Variables

- **Type #1:** the unit type to convert from.
- **Amount #1:** the amount of the first type.
- **Type #2:** the unit type to convert to.
- **Amount #2:** the amount of the second type.

Example

If the temperature is 85 degrees Fahrenheit, what is the equivalent in degrees Celsius?

Variable	Enter	Comments
Tap the Clear… button		Sets the display to its default values
Type #1	Fahrenheit	Choose from the list
Amount #1	85	
Type #2	Celsius	Choose from the list

Select "?" in the Amount #2 row to calculate (may calculate automatically). It is 29.4 degrees Celsius.

4.5.4.5 Time

This template is for Time calculations. It is in the Calendar category. For information on accessing this template, see the Templates : Accessing section. For more on using templates in general, see the Templates : Using the Templates section.

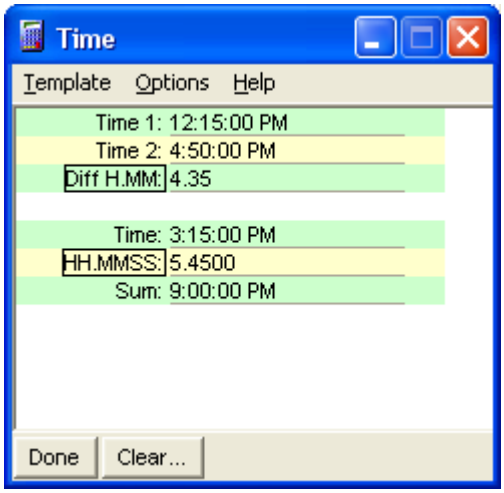
Description

Use this template to calculate the difference between two times or sum two times. Times are displayed in either standard (am/pm) or 24-hour format, depending on the localized time format set in the device's system settings (see your device manual for details).

Variables

The top section of the template performs time difference calculations:

- **Time 1:** the beginning time set in increments of 5 minutes.
- **Time 2:** the ending time set in increments of 5 minutes.
- **Diff H.MM/Diff Hrs:** the difference between the two times. "Diff H.MM" displays in hour-minute format (5 hrs, 45mins would display as 5.45). "Diff Hrs" displays as fraction of an hour.



The bottom section performs time addition calculations:

- **Time:** starting time.
- **HH.MMSS/HrsFrac:** time duration. "HH.MMSS" displays time in hour, minute, second format (4hrs, 7mins, 30sec would display as 4.0730). "Hrs.Frac" displays time as a fraction of an hour.
- **Sum:** the end time.

Example

Time Difference: A project began at 12:15pm and concluded at 4:50pm. How much time was spent on the project?

Variable	Enter	Comments
Tap the Clear… button		Sets the display to its default values
Time 1	12:15pm	
Time 2	4:50pm	

Make sure Diff Hrs is showing instead of Diff H.MM. Select "?" in the Diff Hrs row to calculate. This project took 4.5833 hours (4 hours, 35 minutes).

Time Sum: If you start driving at 3:15 pm and the trip will take 5hrs, 45mins, what time will you arrive?

Variable	Enter	Comments
Tap the Clear… button		Sets the display to its default values
Time	3:15 pm	
H.MMSS	5.45	

Select "?" in the Sum row to calculate. You will arrive at 9:00pm.

4.5.4.6 Time Value of Money

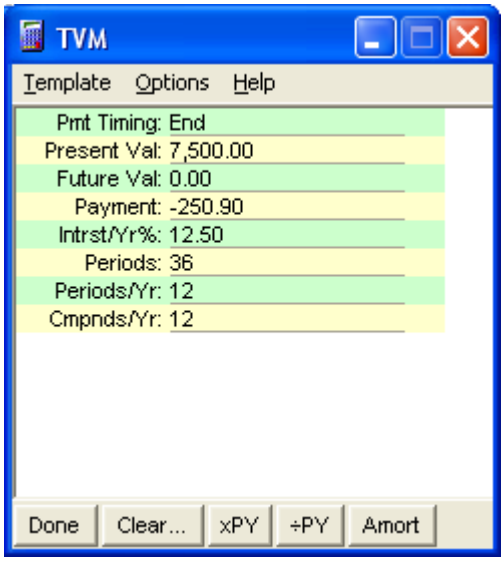
This template is for TVM (Time Value of Money) calculations. See the Included Templates section for category information.

Description

Time value of money is the process of earning compound interest over a period of time. Compound interest problems assume that the interest earned also earns interest. Computations such as loans, leases, mortgages, annuities, and savings accounts are examples of compound interest problems.

In time value problems, positive and negative numbers have different meanings: positive numbers are inflows of cash (cash received) while negative numbers are outflows (cash paid). A car loan, for instance, may have a positive present value (because money was received from the loan company) but will have a negative payment amount, since this is money that will be paid back to the loan company. See the Understanding Cash Flows section for a pictorial explanation

of inflows and outflows.



Variables

- **Pmt Timing:** the payment timing. Payments occur at the beginning or end of the period. Payments made at the beginning of the period are called Annuity Due. Most leases are this kind. A payment made at the end of the period is called an Ordinary Annuity. Most loans are this kind.
- **Present Val:** the present value.
- **Future Val:** the future value.
- **Payment:** payment amount per period.
- **Intrst/Yr%:** interest per year as a percentage. For example, 8.25% interest should be entered as "8.25".
- **Periods:** number of total periods. This number is the number of years and months times the periods per year. For example, if the loan is 4 years with 12 payments per year (monthly payments), periods should be 48 (4 x 12).
- **Periods/Yr:** the number of payment periods per year. For example, if payments are made quarterly, periods per year should be 4.
- **Cmpnds/Yr:** the number of interest compounding periods per year. Most of the time, compounding periods per year should equal payment periods per year. For example, if payments are made monthly and interest is compounded monthly, compounding periods per year and periods per year should both be 12.

Buttons

- **x:** quick set button for the number of periods. This button multiplies the value in periods by the value in periods per year. For example, to convert 10 years at 12 periods per year to periods, enter 10 in periods, 12 in periods per year, and select the x button.
- **÷:** quick set button for the number of periods. This button divides the value in periods by the value in periods per year. For example, if periods is 60 with periods per year equal to 12, discovering that it is equal to five years can be done easily by selecting the ÷ button.
- **Amort**: access to the amortization screen. See the Amortization section for more information. If no data is entered in the TVM template, the amortization screen will not appear.

Examples

Car Loan: When purchasing a new car, the auto dealer has offered a 12.5% interest rate over 36 months on a $7,500 loan. What will be the monthly payment?

Variable	Enter	Comments
Tap the Clear… button		Sets the display to its default values
Pmt Timing	End	Loan pmt's are at the end of the period. Choose from the list
Present Val	7500	
Intrst/Yr%	12.5	
Periods	36	3 years at 12 periods per year
Periods/Yr	12	
Cmpnds/Yr	12	

Select "?" in the Payment row to calculate. The payment will be –250.90 per month. It is negative because it is a cash outflow.

Car Loan, Amortization 1: How much interest was paid for the first payment? (Assumes you are currently in the TVM template.)

Button/Control	Enter	Comments
Amort		Goes to the amortization template
Interest		Choose Interest from the data display 1 or 2 pop-up list

Interest for the first period is –78.13. This is negative because it is part of the payment, which is a cash outflow.

Car Loan, Amortization 2: How much principal was paid for the first year if the car was purchased in January? (Assumes you are currently in the TVM template.)

Variable	Enter	Comments
Amort		Goes to the amortization template
Period		Select the amortization view in the top, right-hand corner.
Beg Period	1	
End Period	12	

Compute by selecting "?" next to Beg Balance. The principal paid for the first year is –2196.29. This value is negative because it is part of the payment, which is a cash outflow.

Retirement Annuity: With 35 years until retirement and $15,000 in the bank, it is time to think about savings. How much would have to be put aside at the beginning of each month to reach $2.5 million if an interest rate of 10% can be expected.

Variable	Enter	Comments
Tap the Clear… button		Sets the display to its default values
Pmt Timing	Begin	Choose from the list
Present Val	-15,000	Negative because cash out of hand
Future Val	2,500,000	Positive because future cash inflow
Intrst/Yr%	10.0	
Periods	420	35 years x 12 periods per year
Periods/Yr	12	
Cmpnds/Yr	12	

Select "?" in the Payment row to calculate. The payment amount is –525.15 per month. It is negative because it is a cash outflow.

Savings Account: With $3,000 in a savings account and 3.75% interest, how many months does it take to reach $4,000?

Variable	Enter	Comments
Tap the Clear… button		Sets the display to its default values
Pmt Timing	End	Choose from the list
Present Val	-3,000	Negative because cash deposit (outflow) to open account
Future Val	4,000	
Payment	0	
Intrst/Yr%	3.75	
Periods/Yr	12	
Cmpnds/Yr	12	

Select "?" in the Periods row to calculate. To reach $4,000, it will take 92.20 periods (or 92.20 ÷ 12 = 7.68 years).

Home Mortgage: You have decided to buy a house but you only have $900 to spend each month on a 30-year mortgage. The bank has quoted an interest rate of 8.75%. What is the maximum purchase price?

Variable	Enter	Comments
Tap the Clear… button		Sets the display to its default values
Pmt Timing	End	Loans payments at the end of the period. Choose from the list
Future Val	0	
Payment	-900	Negative because cash outflow
Intrst/Yr%	8.75	
Periods	360	30 years at 12 periods per year
Periods/Yr	12	
Cmpnds/Yr	12	

Select "?" in the Present Val row to calculate. You can afford a home with a price of $114,401.87.

Mortgage with a Balloon Payment: (Continued from Home Mortgage) You realize that you will only own the house for about 5 years and then sell it. How much will the balloon payment (the repayment to the bank) be?

Variable	Enter	Comments
Periods	60	5 years at 12 periods per year

Select "?" in the Future Val row to calculate. The balloon payment will be $109,469.92 after five years.

Canadian Mortgage: Canadian mortgages compound interest twice per year instead of monthly. What is the monthly payment to fully amortize a 30-year, $80,000 Canadian mortgage if the interest rate is 12%?

Variable	Enter	Comments
Tap the Clear… button		Sets the display to its default values
Pmt Timing	End	Loans payments at the end of the period. Choose from the list
Present Val	80,000	Positive because cash inflow
Future Val	0	
Intrst/Yr%	12.00	
Periods	360	30 years at 12 periods per year
Periods/Yr	12	
Cmpnds/Yr	2	

Select "?" in the Payment row to calculate. The payment will be –$805.11. It is negative because it is a cash outflow.

Bi-Weekly Mortgage Payments: A buyer is considering a $100,000 home loan with monthly payments, an annual interest rate of 9% and a term of 30 years. Instead of making monthly payments, the buyer realizes that he can build equity faster by making bi-weekly payments (every two weeks). How long will it take to pay off the loan?

Part 1: Calculate the monthly payment

Variable	Enter	Comments
Tap the Clear… button		Sets the display to its default values
Pmt Timing	End	Choose from the list
Present Val	100,000	
Future Val	0	
Intrst/Yr%	9.00	
Periods	360	30 years at 12 periods per year
Periods/Yr	12	
Cmpnds/Yr	12	

Select "?" in the Payment row to calculate. Calculating shows payment equal to –$804.62. It is negative because it is a cash outflow.

Part 2: Periods when making bi-weekly payments (continued)

Variable	Enter	Comments
Payment	-402.31	Recall payment in the input screen and divide it by 2
Periods/Yr	26	Bi-weekly payments mean 26 per year
Cmpnds/Yr	12	Still compounding interest monthly

Select "?" in the Periods row to calculate. Calculating shows periods equal to 567.40 periods (567.40 ÷ 26 = 21.82 years).

APR of a Loan with Fees: The Annual Percentage Rate (APR) is the interest rate when fees are included with the

mortgage amount. Because the fees increase the cost of the loan, the effective interest rate on the borrowed amount is higher. For example, a borrower is charged two points for the issuance of a mortgage (one point is equal to 1% of the mortgage amount). If the mortgage amount is $60,000 for 30 years with an interest rate of 11.5%, what is the APR?

Part 1: Calculate the actual monthly payment

Variable	Enter	Comments
Tap the Clear… button		Sets the display to its default values
Pmt Timing	End	Choose from the list
Present Val	60,000	
Future Val	0	
Intrst/Yr%	11.5	
Periods	360	30 years at 12 periods per year
Periods/Yr	12	
Cmpnds/Yr	12	

Select "?" in the Payment row to calculate. Calculating shows payment equal to –$594.17. It is negative because it is a cash outflow.

Part 2: Calculate the APR (continued)

Variable	Enter	Comments
Present Val	58,800	The loan amount less 2% in fees. Calculate in the input screen with 60000 [x] .02 [=] [+/–] [+] 60000 [=]

Select "?" in the Intrst/Yr% row to calculate. Calculating shows interest per year equal to 11.76%.

Present Value of a Lease with Advance Payments and an Option to Buy: With a lease, often there is an amount to be paid up-front and an option to buy at the back-end. A company is leasing a machine for 4 years. Monthly payments are $2,400; an additional $2,400 payment at the beginning of the leasing period replaces the final payment. The leasing agreement includes an option to buy the machine for $15,000 at the end of the leasing period. What is the capitalized value of the lease, assuming that the interest rate paid to borrow the funds is 18% compounded monthly?

Part 1: Find the present value of the payments

Variable	Enter	Comments
Tap the Clear… button		Sets the display to its default values
Pmt Timing	Beg	Choose from the list
Future Val	0	
Payment	–2,400	
Intrst/Yr%	18.00	
Periods	47	4 years at 12 per year less 1 advance payment
Periods/Yr	12	
Cmpnds/Yr	12	

Select "?" in the Present Val row to calculate. Calculating shows present value equal to $81,735.58. Recall this to the pop-up calculator and save it to memory.

Part 2: Present Value of the buy option (continued)

Variable	Enter	Comments
Future Val	–15,000	
Payment	0	
Periods	48	

Select "?" in the Present Val row to calculate. Calculating shows present value equal to $7,340.43.

Part 3: Calculate (continued)

Recall the present value to the pop-up calculator by selecting the Present Value's amount. Add it the present value of the payments stored in memory and $2,400 for the advanced payment. The answer is $91,476.00.

4.5.4.6.1 Amortizations

The amortization screen displays period-by-period information. Included information is beginning and ending balances, payment, interest and principal amounts. There are two amortization views: table and period.

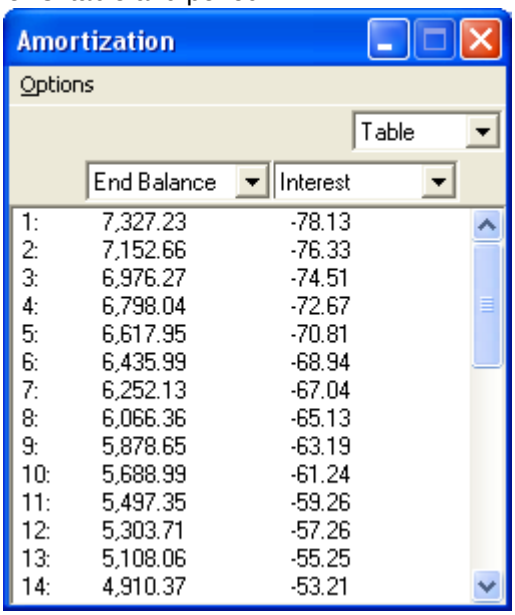


Table View

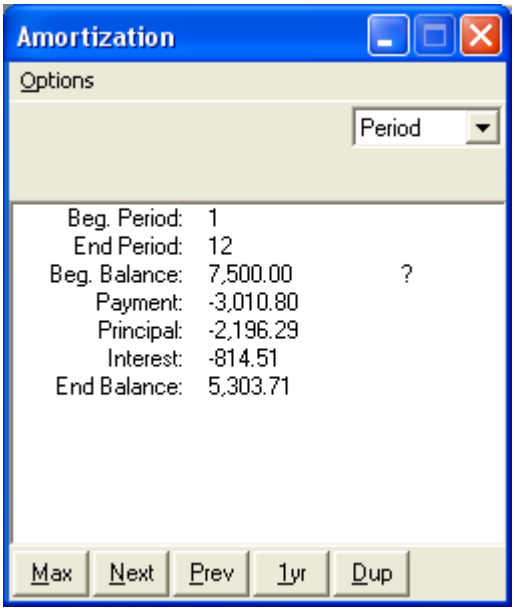
The table view, set by choosing "Table" from the pop-up list in the top right corner, displays period-by-period information.

The display, from left-to-right, shows the period being displayed, data number one, and data number two. Either of the data sets can display beginning or ending balance, payment, principal, or interest. Select one by tapping on the pop-up list above each column.

Scroll through the list with either the scroll arrows or scroll buttons. Navigate to a specific period by selecting "Go To", entering a period, and selecting the "✓" (save button).

Period View

The period view can be selected by choosing "Period" from the pop-up list in the top, right-hand corner.



To calculate the values over the range of periods, enter a period for Beg Period, one for End Period, and select "?". The table will display information for Beg Balance, End Balance, Payment, Principal, and Interest over that range.

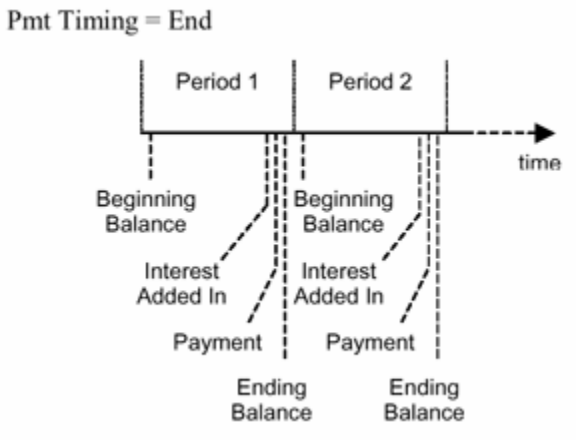
Additionally, four buttons are available at the bottom of the screen next to "Done". These buttons are used to quickly enter beginning and ending period values and perform computations:

- **Max:** sets the ending period to the maximum number of periods.
- **Next:** moves to the next set of beginning and ending periods. For example, with both set to one, selecting next moves the beginning and ending period to two.
- **1yr:** calculates one year from the beginning period.
- **Dup:** duplicates the beginning period in the ending period.

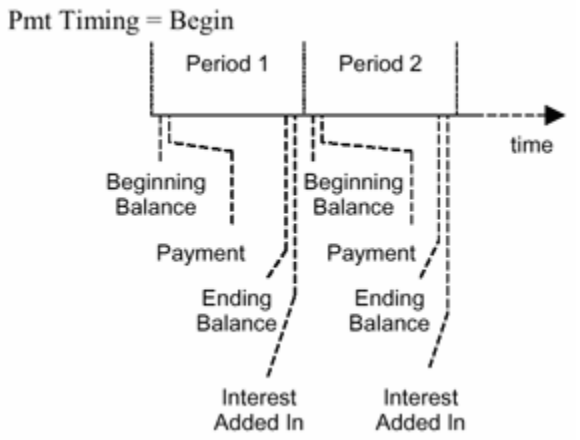
Payment Timing Issues

Whether payments are made at the beginning or end of the period impacts how amortization information is calculated. This impacts the values shown in the amortization table in a way best illustrated by time line diagrams:

When payments are made at the end of the period (common for loans and mortgages), interest is accrued just before the payment is due. Note that the Ending Balance will always be the same as the Beginning Balance for the following payment period.



When payments are made at the beginning of the period (common for leases), interest is accrued after the end of the period. Note that the Ending Balance of a period will be less than the Beginning Balance of the next period by the amount of interest that accrued during that period.

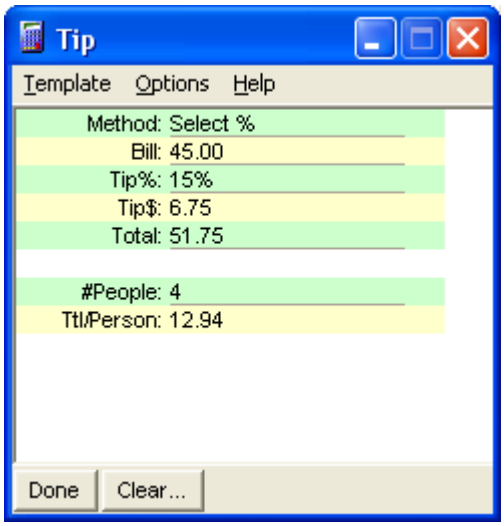


4.5.4.7 Tip

This template is for Tip and bill splitting calculations. It is in the Business category. For information on accessing this template, see the Templates : Accessing section. For more on using templates in general, see the Templates : Using the Templates section.

Variables

- **Method:** the calculation method. Choose "Select%" to select a tip percentage from a list, choose "Enter%" to enter the tip's percentage as a number or choose "Enter$" enter the tip's amount.
- **Bill:** the bill amount before tip.
- **Tip%:** tip amount entered or calculated as a percentage.
- **Tip$:** tip amount entered or calculated as an amount.
- **Total:** the total amount including the tip.
- **#People:** the number of people paying for the meal.
- **Ttl/Person:** the total per person, split evenly among the people paying for the meal.



The tip computation calculates from the top, down. In other words, to calculate the tip amount, it will first attempt to use Bill, then Total, and finally Ttl/Person to calculate. It knows to use a value when the variable is not zero. If Bill is 0, for example, it will use Total.

Example

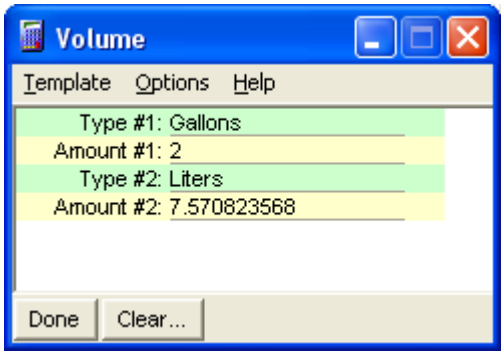
A lunch bill for four friends is $45. With a 15% tip, what is the total bill amount how much should each person contribute?

Variable	Enter	Comments
Tap the Clear… button		Sets the display to its default values
Bill	45	
Tip%	15	
#People	4	

Select "?" in the Tip$ and Total rows to calculate the tip amount and total amount, respectively, and select "?" in the Ttl/Person row to calculate the amount per person. The total is $51.75 and each person should contribute $12.94.

4.5.4.8 Volume

This template is for Volume conversions. It is in the Convert category. For information on accessing this template, see the Templates : Accessing section. For more on using templates in general, see the Templates : Using the Templates section.



Variables

- **Type #1:** the unit type to convert from.
- **Amount #1:** the amount of the first type.
- **Type #2:** the unit type to convert to.
- **Amount #2:** the amount of the second type.

Example

For athletics, the coaches suggest drinking 2 gallons of water a day. How much water is this in Liters?

Variable	Enter	Comments
Tap the Clear… button		Sets the display to its default values
Type #1	Gallons	Choose from the list
Amount #1	2	
Type #2	Liters	Choose from the list

Select "?" in the Amount #2 row to calculate (may calculate automatically). The answer is 7.6 Liters of water.

4.6 Creating Templates

To create a template from the main calculator, select the Template button (third from left) and choose "New Template" at the bottom of the list. To create a template from My Templates, select "New". Once created, these templates are accessed the same as templates that are included with the software. See Templates : Accessing for more information.

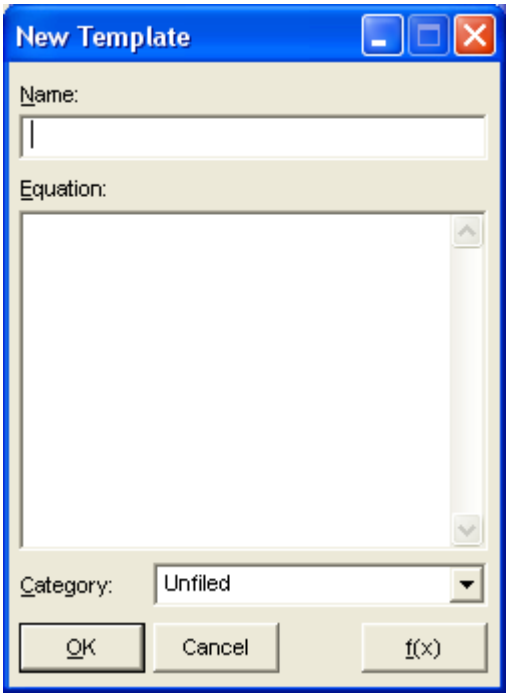
A. Name: template's name.

B. Equation: template's equation. See the Using the Solver section for more information about writing solver equations. Single equation templates are available. See the Solver Limitations section for more information.

C. Category: templates are sorted into categories. Choose a category from the list or select "New Category" to create a new one. Categories only exist as long as there are templates in them. To delete a category, move all the templates in that category to another category or delete all templates in that category.

D. f(x): displays the list of functions organized by category. This is the same as the main calculator's list except function "solving" is available in category Boolean (bool) and category MEM is added (memory store, recall and clear options). Template creation "=" is different than "=" in the main calculator. See the Using the Solver category for more information. See the Functions section for more on each available function. See the Using the Calculator : Memory & Storage : Memory Locations section for more on recall and store.

E. Buttons: "OK" saves changes while "Cancel" deletes changes, returning to My Data. To enter notes about the template, select "Notes".



4.6.1 Using the Solver

An equation is in the form a = b where a and b are combinations of variables, constants, operators and functions. The following equation will be used for reference:

Area = sin(Width/45) * Length

An equation is made of four components:

Variables: the names of items that are either stored or calculated. In the example above, Area, Width and Length are variables.

These variables must consist of letters (capital or lower case 'a' through 'z') and numbers (0 through 9) with a maximum of 11 characters. The variable cannot start with a number. Note that the percentage symbol (%) is a mathematical symbol and cannot be used in the name of a variable. (The built-in templates are able to use a label to represent the variable in the template view and the label is not subject to the same restrictions as a variable's name.)

Constants: these are values that do not change. In the example above, 45 is a constant.

Do not use digit separators (such as commas or spaces). For decimal separators, use the setting defined in the system's Preferences and indicated as the decimal separator button (either point or comma). Use the keypad to enter these.

Operators: mathematical symbols such as plus, minus, times and divide. In the example above, both multiple (*) and divide (/) are operators.

Functions: allows for more advanced mathematical capabilities, which are built into the calculator. In the example above, sine (sin) is a function.

Select the "f(x)" button on the keypad to access a list of functions sorted by category. See the Function section for more information about each.

Additional notes on entering equations:

- The solver follows order of operations precedence. To override order of operations or in cases where order of operations is uncertain, use parentheses in the formula.

- Spaces are ignored. Often, when the equation is strung together on the screen without any spaces, it is difficult to see. Use spaces to help view equations.
- There is no implied multiplication. If an equation shows "z (1 + h)", that needs to be entered as "z * (1 + h)".
- Often an either/or situation exists when performing a calculation. "If" statements are used to express these relationships.

4.6.2 How the Solver Works

There are two kinds of solving: iterative and symbolic. Infinity Softworks' solver uses an iterative approach.

An iterative method attempts to balance an equation and determine an answer, relying on a minimum and maximum guess to "bracket" the answer. An equation is said to be in balance when, tabulated, the value to the left of the equals sign is the same as the value to the right of the equals sign.

In its simplest form, an iterative solver determines a mid-point between a maximum and minimum guess and evaluates the equation at all three points. It then decides which two points the equation is between – the mid-point and minimum guess or mid-point and maximum guess – and calculates a new mid-point based on those two points. It continues this cycle until it "guesses" the right answer. Although the solver is more advanced than this, it is similar in nature.

Generally, calculations are completed in less than a second. If the calculation is more complicated, however, a Computing dialog will appear. Guesses made by the solver flash on the screen. If the "Cancel" button is selected, the answer will return as the last guess.

To speed execution and increase the likelihood of an answer, change the max and min range settings in the Variable Preferences. See the Using the Templates : Variable Preferences section for more information. In some cases, the solver can calculate an answer directly without iterating to an answer. In this case, max and min range settings are not available.

4.6.3 Solver Limitations

Certain functionality offered with included templates is not available with solver created ones. Commonly requested one are listed here:

- **Data entry:** only numbers (Booleans, Integers and Floating Point Numbers) can be entered into a template, although dates (dd.mmyyyy) and times (hh.mmssmmm) can be entered in number formats. Other data types can be used in the equation but cannot be entered.
- **Multiple equations:** templates are designed to work with single equations only, although multiple equations can be derived by using choose, if and solving in the equation. See the Templates : Creating Templates : Examples section for more information.
- **Buttons:** there is no mechanism for creating buttons next to Done and Clear.
- **Pop-up Lists:** there is no mechanism for creating pop-up lists.

4.6.4 Examples

This section includes examples for creating custom templates. From time to time, Infinity Softworks posts supplemental materials on its web site. See product support at www.infinitysw.com/finance for additional examples.

4.6.4.1 Inflation

This example demonstrates template creation basics.

The Example

The equation for inflation is:

FutureVal = PresentVal * ((1+Inflation/100)^Years)

where

- FutureVal is the future amount.
- PresentVal is the present amount.
- Inflation is the rate entered as a percentage (4 instead of .04).
- Years is the number of years between PresentVal and FutureVal.

The problem to calculate: if a pool costs $5,000, what is the expected cost 5 years from now if the inflation rate is 4%?

Creating the Template

1. Launch the software. The main calculator should be visible.

2. Select the Template button. The Template button is third from the left across the top of the main calculator. The Template List should be visible.

3. Select "New Template" near the bottom of the list.

4. Enter "Inflation" (no quotes) for the name.

5. Enter "FutureVal = PresentVal * ((1+Inflation/100)^Years)" (no quotes) for the equation.
   - equals (=) is on the right side of the keypad.
   - power (^) is available by selecting "f(x)" then math then "y^x".

6. Select "OK" to finish. The Inflation template should be visible.

Calculate Purchasing Power

7. Enter $5,000 for the present value.
   - select 0.00 next to PresentVal.
   - enter 5000 in the pop-up calculator using the keypad.
   - select the save ("✓") button.

8. Enter 4% inflation.
   - select 0 next to Inflation.
   - enter 4 in the pop-up calculator using the keypad.
   - select the save ("✓") button.

9. Enter 5 years.
   - select 0 next to Years.
   - enter 5 in the pop-up calculator using the keypad.
   - select the save ("✓") button.

10. Calculate the future value.
    - select the "?" button on the same line as FutureVal.
    - the cost after 5 years is $6,083.26.

For more on formatting variables, see the Templates : Using the Templates : Variable Preferences section. This template and others are available for free download from Infinity Softworks' web site: www.infinitysw.com/finance.

4.6.4.2 Constant Acceleration

This example demonstrates template creation basics.

The Example

The equation for constant acceleration is:

Velocity1^2 = Velocity0^2 + 2 * Accelrtn * Distance

where

- Velocity1 is the final velocity.
- Velocity0 is the initial velocity.
- Accelrtn is the speed of acceleration. Negative denotes deceleration.
- Distance is the distance between Velocity1 and Velocity0.

The problem to calculate: what is the stopping distance for a car traveling 30 meters per second but decelerating 5 meters per second squared?

Creating the Template

1. Launch the software. The main calculator should be visible.

2. Select the Template button. The Template button is third from the left across the top of the main calculator. The Template List should be visible.

3. Select "New Template" near the bottom of the list.

4. Enter "Const Acclrtn" (no quotes) for the name.

5. Enter "Velocity1^2 = Velocity0^2 + 2 * Accelrtn * Distance" (no quotes) for the equation.
   - equals (=) is on the right side of the keypad.
   - power (^) is available by selecting f(x) then math then y^x.

6. Select "OK" to finish. The constant acceleration template should be visible.

Calculate Distance

7. Leave Velocity1 as 0.

8. Enter 30 m/s for Velocity0.
   - select 0 next to Velocity0.
   - enter 30 in the pop-up calculator using the keypad.
   - select the save ("✓") button.

9. Enter -5 m/s^2 for acceleration.
   - select 0 next to Accelrtn.
   - enter -5 in the pop-up calculator using the keypad.
   - select the save ("✓") button.

10. Calculate the distance.
    - select the "?" button on the same line as Distance.
    - the car will stop after 90 meters.

For more on formatting variables, see the Templates : Using the Templates : Variable Preferences section. This template and others are available for free download from Infinity Softworks' web site: www.infinitysw.com/finance.

4.6.4.3 Home Loan

This example demonstrates use of the financial functions. This template requires the Finance Library. The finance library is installed if the function list shows the category finance at the bottom. If it does not appear, reinstall the finance library.

The Example

The equation for calculating a mortgage is:

-Payment = tvmpmt(Years*12; IntRate; Mortgage; 0; 12; 12; 0)

where

- Payment is the monthly house payment.
- Years is the number of years to pay off the loan (multiply by 12 to get total number of periods).
- IntRate is the rate entered as a percentage (4 instead of .04).
- Mortgage is the loan amount.

(0; 12; 12; 0 are constants included in the function call. These are added to simplify data entry. The first 0 assumes the mortgage is paid down to 0. The first 12 assumes monthly payments. The second 12 assumes interest compounds monthly. The second 0 assumes payments are made at the end of the period, common for loans and mortgages.)

The problem to calculate: what is the monthly payment to pay off a $300,000 mortgage at 6.75% interest over 30 years?

Creating the Template

1. Launch the software. The main calculator should be visible.

2. Select the Template button. The Template button is third from the left across the top of the main calculator. The Template List should be visible.

3. Select "New Template" near the bottom of the list.

4. Enter "Mortgage" (no quotes) for the name.

5. Enter "-Payment = tvmpmt(Years*12; IntRate; Mortgage; 0; 12; 12; 0)" (no quotes) for the equation.
   - negative and minus are the same. It is on the right side of the keypad.
   - equals (=) is on the right side of the keypad.
   - TVM payment (tvmpmt) is available by selecting f(x) then finance at the bottom of the list. If the category finance is not available, the finance library may not be installed. Reinstall the finance library to complete this problem.

6. Select "OK" to finish. The mortgage template should be visible.

Calculate Monthly Payment

7. Enter 30 years.
   - select 0 next to Years.
   - enter 30 in the pop-up calculator using the keypad.
   - select the save ("✓") button.

8. Enter 6.75% interest rate.
   - select 0 next to IntRate.
   - enter 6.75 in the pop-up calculator using the keypad.
   - select the save ("✓") button.

9. Enter a $300,000 mortgage.
   - select 0 next to Mortgage.
   - enter 300000 in the pop-up calculator using the keypad.
   - select the save ("✓") button.

10. Calculate the monthly payment.
    - select the "?" button on the same line as Payment.
    - the payment is $1,945.79 per month.

For more on formatting variables, see the Templates : Using the Templates : Variable Preferences section. This template and others are available for free download from Infinity Softworks' web site: www.infinitysw.com/finance.

4.6.4.4 "IF" Statements

If statements are used for conditional situations. If statements can be used by themselves or nested, meaning a second if statement is used within a first (the example below uses a nested if statement).

The Example

In this example, a profit sharing formula has three levels:

- If net income is less than or equal to $1 million, there is no profit sharing.
- If net income is greater than $1 million but less than or equal to $5 million, profit sharing is 2% of monthly pay.
- If net income is greater than $5 million, profit sharing is 4% of monthly pay.

With a monthly base of $3000, what is the profit sharing amount if the company's net income is $700,000, $2 million and $10 million?

The equation for calculating this profit sharing formula is:

NetPay = BasePay + if(NetIncome <= 1000000; 0; if(NetIncome > 1000000 && NetIncome <= 5000000; BasePay * .02; BasePay * .04))

where

- NetPay is the final, monthly net pay including base pay and profit sharing.
- BasePay is the monthly base pay.
- NetIncome is the net income earned by the company.

The format for if statements is if(conditional true; do this; otherwise do this). To break down the equation:

- The first if statement says if net income is less than or equal to (<=) 1,000,000, add 0 otherwise do the second if statement.
- The second if statement says if net income is greater than (>) 1,000,000 and (&&) net income is less than or equal to (<=) $5,000,000, then add in 2% of the base pay. If it doesn't meet this condition, than net income must be larger since we took care of all other conditions. Add in 4% of base pay instead.
- Note that nested if statements read from left to right. If the first criteria is true, the solver will not continue to the false statement. Because of that, the formula could be written as: NetPay = BasePay + if(NetIncome <= 1000000; 0; if(NetIncome <= 5000000; BasePay * .02; BasePay * .04)) leaving out "NetIncome > 1000000 &&" in the second, nested if statement.

Creating the Template

1. Launch the software. The main calculator should be visible.

2. Select the Template button. The Template button is third from the left across the top of the main calculator. The Template List should be visible.

3. Select "New Template" near the bottom of the list.

4. Enter "Profit Sharing" (no quotes) for the name.

5. Enter "NetPay = BasePay + if(NetIncome <= 1000000; 0; if(NetIncome > 1000000 && NetIncome <= 5000000; BasePay * .02; BasePay * .04))" (no quotes) for the equation.
   - equals (=) is on the right side of the keypad.
   - parentheses ( ) are on the right side of the keypad next to multiply.
   - if, less then or equal to (<=), and greater than (>) are available by selecting "f(x)" then "bool".
   - Semi-colon (;) is on the left side of the keypad below "f(x)".

6. Select "OK" to finish. The profit sharing template should be visible.

Calculate NetPay (Net Income is $700,000)

7. Enter $3,000 for base pay.
   - select 0 next to BasePay.
   - enter 3000 in the pop-up calculator using the keypad.
   - select the save ("✓") button.

8. Enter $700,000 for net income.
   - select 0 next to NetPay.
   - enter 700000 in the pop-up calculator using the keypad.
   - select the save ("✓") button.

9. Calculate the monthly payment.
   - select the "?" button on the same line as NetPay.
   - the net pay is $3,000.

Calculate NetPay (Net Income is $2,000,000)

10. Enter $2,000,000 for net income.
    - select 700,000 next to NetPay.
    - enter 2000000 in the pop-up calculator using the keypad.
    - select the save ("✓") button.
11. Calculate the monthly payment.
    - select the "?" button on the same line as NetPay.
    - the net pay is $3,060.

Calculate NetPay (Net Income is $10,000,000)

12. Enter $10,000,000 for net income.
    - select 2,000,000 next to NetPay.
    - enter 10000000 in the pop-up calculator using the keypad.
    - select the save ("✓") button.
13. Calculate the monthly payment.
    - select the "?" button on the same line as NetPay.
    - the net pay is $3,120.

For more on formatting variables, see the Templates : Using the Templates : Variable Preferences section. This template and others are available for free download from Infinity Softworks' web site: www.infinitysw.com/finance.

4.6.4.5 "Solving" Statements

Solving statements are used in connection with if statements to offer multi-equation calculation capabilities.

The Example

This example calculates both the final price of a product with sales tax and the sale tax itself. The equation for calculating the final price is:

FinalPrice = Cost * (1+TaxRate%)
or
0 = Cost * (1+TaxRate%) - FinalPrice

The equation for calculating the tax amount is:

TaxAmt = Cost * TaxRate%
or
0 = Cost * TaxRate% - TaxAmt

where

- Cost is the cost of the good or service.
- TaxRate is the tax rate expressed as a percentage (4 instead of .04).
- FinalPrice is the final price of the good or service with sales taxes.
- TaxAmt is the amount of the sales tax.

On a $29.99 purchase in a state with 7.25% sales tax, what is the final price of the product and how much is paid in sales tax?

In this case, the formula is:

0 = if(solving() == "Cost" || solving() == "FinalPrice"; Cost * (1 + TaxRate%) - FinalPrice;
Cost * TaxRate% - TaxAmt)

The format for if/solving statements is if(solving() == "variable"; do this; otherwise do this). To break down the equation:

- (solving() == "Cost" || solving() == "FinalPrice") says if the "?" button next to Cost or (||) FinalPrice is selected, perform the first calculation otherwise perform the second calculation.
- The parentheses next to solving are required, otherwise the solver thinks it is variable "solving".

- Notice the double equals sign (==), which is different than the single version (=). The single equals is assignment while the double equals is comparison.
- Because two equations are used in the calculation, it is not possible to put a variable on the left side of the equals sign. In this case "0 =" is used and the variables being calculated (FinalPrice and TaxAmt) are moved to the right side as a part of the equation. Leaving "0 =" out of the equation causes calculation problems.

Creating the Template

1. Launch the software. The main calculator should be visible.

2. Select the Template button. The Template button is third from the left across the top of the main calculator. The Template List should be visible.

3. Select "New Template" near the bottom of the list.

4. Enter "Sales Tax" (no quotes) for the name.

5. Enter "0 = if(solving() == "Cost" || solving() == "FinalPrice"; Cost * (1 + TaxRate%) - FinalPrice; Cost * TaxRate% - TaxAmt)" (no quotes) for the equation.
   - equals (=) is on the right side of the keypad next to plus.
   - if, solving(), or (||) and comparison equals (==) are available by selecting "f(x)" then "bool".
   - Quotes (" ") are on the right side of the keypad next to equals.
   - Semi-colon (;) is on the left side of the keypad below "f(x)".
   - Percent (% or %x) is available by selecting "f(x)" then "math". This is the same as TaxRate/100.
   - parentheses ( ) are on the right side of the keypad next to multiply.

6. Select "OK" to finish. The sales tax template should be visible.

7. Turn off auto-solve in the template preferences.
   - select the menu button (below the Applications button) or, on some devices, select the title bar.
   - choose the Options menu.
   - select "Template Prefs".
   - uncheck Auto-Compute.
   - select "OK". The sales tax template should be visible.

Calculate Final Price with Sales Tax

8. Enter $29.99 for cost.
   - select 0 next to Cost.
   - enter 29.99 in the pop-up calculator using the keypad.
   - select the save ("✓") button.

9. Enter 7.25% for sales tax rate.
   - select 0 next to TaxRate.
   - enter 7.25 in the pop-up calculator using the keypad.
   - select the save ("✓") button.

10. Calculate the final price.
    - select the "?" button on the same line as FinalPrice.
    - the final price, including sales tax, is $32.16.

Calculate the Tax Amount

11. Calculate the sales tax amount.
    - select the "?" button on the same line as TaxAmt.
    - the net pay is $2.17.

For more on formatting variables, see the Templates : Using the Templates : Variable Preferences section. This template and others are available for free download from Infinity Softworks' web site: www.infinitysw.com/finance.

4.6.4.6 Multiple Answers

Some equations offer more than one answer to a problem. If statements and solving make the problem possible. Preferences also need to be adjusted to make sure calculations are performed in the appropriate range.

The Example

This example calculates both the positive and negative values of x when y is equal to $x^2 - 3$.

In this case, the formula is:

0 = if(solving() == "PosX"; -y + PosX^2 - 3; -y + NegX^2 - 3)

See the "IF" Statement and "Solving" Statement examples for more on using these functions in solver equations.

Creating the Template

1. Launch the software. The main calculator should be visible.

2. Select the Template button. The Template button is third from the left across the top of the main calculator. The Template List should be visible.

3. Select "New Template" near the bottom of the list.

4. Enter "Quadratic" (no quotes) for the name.

5. Enter "0 = if(solving() == "PosX"; -y + PosX^2 - 3; -y + NegX^2 - 3)" (no quotes) for the equation.
   - equals (=) is on the right side of the keypad next to plus.
   - if, solving(), and comparison equals (==) are available by selecting "f(x)" then "bool".
   - Quotes (" ") are on the right side of the keypad next to equals.
   - Semi-colon (;) is on the left side of the keypad below "f(x)".

6. Select "OK" to finish. The quadratic template should be visible.

7. Turn off auto-solve in the template preferences.
   - select the menu button (below the Applications button) or, on some devices, select the title bar.
   - choose the Options menu.
   - select "Template Prefs".
   - uncheck Auto-Compute.
   - select "OK". The sales tax template should be visible.

8. Adjust the range for PosX and NegX. The solver usually finds positive results first. Adjust the range to force the negative answer.
   - select the menu button (below the Applications button) or, on some devices, select the title bar.
   - choose the Options menu.
   - select "Variable Prefs".
   - select "y" next to Variable.
   - choose "PosX".
   - change Range : Min to 0 by selecting the arrow to the right of min and selecting "0".
   - select "PosX" next to Variable.
   - choose "NegX".
   - change Range : Max to 0 by selecting the arrow to the right of max and selecting "0".

Calculate the Positive X Value

9. Enter 9 for y.
   - select 0 next to y.
   - enter 9 in the pop-up calculator using the keypad.
   - select the save ("✓") button.

10. Calculate the positive x value.
   - select the "?" button on the same line as PosX.
   - the positive x value is 3.46.

Calculate the Negative X Value

10. Calculate the negative x value.
- select the "?" button on the same line as NegX.
- the negative x value is -3.46.

For more on formatting variables, see the Templates : Using the Templates : Variable Preferences section. This template and others are available for free download from Infinity Softworks' web site: www.infinitysw.com/finance.

4.6.4.7 Using Data in Multiple Templates

The global preference setting for variables makes it possible to share data between templates.

The Example

This example will utilize the inflation template created earlier.

The problem: summarize average monthly expenses and determine the equivalent amount needed 30 years from now to retire. The monthly expenses to consider:

- Rents including gas, electricity, water and garbage
- Health Insurance
- Groceries
- Automobile Expenses
- Entertainment

To perform the calculation, one template will be created to summarize these expenses. This template will share the total amount with a second, inflation calculation template.

Create the Inflation Template

If the inflation template has already been created, skip to step 7.

1. Launch the software. The main calculator should be visible.

2. Select the Template button. The Template button is third from the left across the top of the main calculator. The Template List should be visible.

3. Select "New Template" near the bottom of the list.

4. Enter "Inflation" (no quotes) for the name.

5. Enter "FutureVal = PresentVal * ((1+Inflation/100)^Years)" (no quotes) for the equation.
- equals (=) is on the right side of the keypad.
- power (^) is available by selecting f(x) then math then y^x.

6. Select "OK" to finish. The Inflation template should be visible.

7. Make PresentVal a global variable so its data can be shared.
- select the menu button (below the Applications button) or, on some devices, select the title bar.
- choose the Options menu.
- select "Variable Prefs".
- select "FutureVal" next to Variable.
- choose "PresentVal".
- check the box next to Global.
- select "OK". The inflation template should be visible.

Creating the Expense Summary Template

8. From the Inflation template, select Done.

9. Select the Template button. The Template button is third from the left across the top of the main calculator. The Template List should be visible.

10. Select "New Template" near the bottom of the list.

11. Enter "EXP Summary" (no quotes) for the name.

12. Enter "Rents + Insurance + Groceries + Auto + ENT = PresentVal" (no quotes) for the equation.
    - equals (=) is on the right side of the keypad.

13. Select "OK" to finish. The Expense Summary template should be visible.

14. Make FutureVal a global variable so its data can be shared.
    - select the menu button (below the Applications button) or, on some devices, select the title bar.
    - choose the Options menu.
    - select "Variable Prefs".
    - select "Rents" next to Variable.
    - choose "PresentVal".
    - check the box next to Global.
    - select "OK". The inflation template should be visible.

Calculate Present Value

15. Enter $750 for the total rents.
    - select 0.00 next to Rents.
    - enter 750 in the pop-up calculator using the keypad.
    - select the save ("✓") button.

16. Enter $400 for medical insurance.
    - select 0 next to Insurance.
    - enter 400 in the pop-up calculator using the keypad.
    - select the save ("✓") button.

17. Enter $200 for groceries.
    - select 0 next to Groceries.
    - enter 200 in the pop-up calculator using the keypad.
    - select the save ("✓") button.

18. Enter $150 for automobile expenses.
    - select 0 next to Groceries.
    - enter 150 in the pop-up calculator using the keypad.
    - select the save ("✓") button.

19. Enter $200 for entertainment.
    - select 0 next to ENT.
    - enter 200 in the pop-up calculator using the keypad.
    - select the save ("✓") button.

20. Calculate the total monthly expenses.
    - select the "?" button on the same line as PresentVal.
    - the total monthly expenses are $1,700.

Calculate the Future Cash Requirement to Retire

21. Select "Done" at the bottom of the Inflation template. The main calculator should be visible.

22. Select the Template button. The Template button is third from the left across the top of the main calculator. The Template List should be visible.

23. Select the "Unfiled" category.

24. Select the Inflation template. The Inflation template should be visible with $1,700 as the PresentVal.

25. Enter 4% for inflation.
    - select 0.00 next to Inflation.

- enter 4 in the pop-up calculator using the keypad.
- select the save ("✓") button.

26. Enter 30 years.
    - select 0.00 next to Years.
    - enter 30 in the pop-up calculator using the keypad.
    - select the save ("✓") button.

27. Calculate the future cash requirements for the same monthly expenses.
    - select the "?" button on the same line as FutureVal.
    - the total monthly expenses are $5,513.78.

For more on formatting variables, see the Templates : Using the Templates : Variable Preferences section. This template and others are available for free download from Infinity Softworks' web site: www.infinitysw.com/finance.

4.7 Additional Templates

Additional templates are available at Infinity Softworks' web site:

www.infinitysw.com/finance

These templates include free additions and pay-for add-ons. Options include:

- Mortgages
- Investing
- Leasing and Lending
- Conversions
- Health
- Fitness
- Home and Shopping
- Area and Volume
- Mathematics
- More!

5 Functions

5.1 Function References

This section includes references to each function available in powerOne Finance. Each input mode offers different functions. See the Using the Calculator : Input Modes section for more information on each input mode. RPN, of operations and chain input modes are included with the software.

Different skins offer different methods for accessing functions. See the Functions : Accessing section for more information.

Each function reference has the following elements from top to bottom:

- **Function**: visual indicator of the function. Some functions are commonly known by a different name. See the Functions : Symbol Chart section for a reference.
- **Description**: text description of the function.
- **Data Types**: data comes in various forms. This element describes which data types are appropriate for the selected function. See the Types of Data section for more information.
- **Category**: functions are sorted into categories within the software. This lists the category where the function can be found.
- **Input Modes**: certain functions can be found in certain input modes. This lists the input modes where the function is available.
- **Examples**: examples for the function. These are organized by input mode.

5.2 Accessing

There are three ways to access a function:

1. Select a button in the main calculator.

2. Select an f(x) button in the main calculator. The f(x) or function button lists all functions by category.

5.3 Symbol Chart

This section outlines common symbols and their equivalents within this software. For more on each function, look for the Function Name in alphabetical order in this section.

Symbol	Function Name
-	Subtraction
.	Decimal Separator
,	Decimal Separator
!=	Not Equal
!x	Not
#	Exclusive Or
##	Exclusive Or
%x	Percent
&&	And
( )	Parentheses
/	Division
÷	Division
;	Semi-Colon
||	Or
+	Addition
+/-	Sign
<	Less Than
<=	Less Than or Equal To
!=	Not Equal
=	Equals
==	Equals
>	Greater Than
>=	Greater than or Equal To
^	Power
←	Backspace
$\sqrt{x}$	Square Root
$\sqrt[3]{x}$	Cubed Root
$\sqrt[y]{x}$	Root
1/x	Reciprocal
10^x	Power of 10
abs	Absolute Value
acos	Arc-Cosine
acosh	Hyperbolic Arc-Cosine
adjDate	Adjust Date
adjTime	Adjust Time

AmEndBal	Amortization, End Balance
AmSumInt	Amortization, Interest Paid
AmSumPrn	Amortization, Principal Paid
angle	Rectangular to Polar Conversion
ArmEndBal	ARM Amortization, End Balance
ArmSumInt	ARM Amortization, Interest Paid
ArmSumPrn	ARM Amortization, Principal Paid
asin	Arc-Sine
asinh	Hyperbolic Arc-Sine
atan	Arc-Tangent
atanh	Hyperbolic Arc-Tangent
BondA	Bond Accrued Interest
BondP	Bond Price
BondY	Bond Yield
cbrt	Cubed Root
CE/C	Clear
ceil	Ceiling
CfoCount	Count
CfoIRR	Internal Rate of Return
CfoMIRR	Modified Internal Rate of Return
CfoNFV	Net Future Value
CfoNPV	Net Present Value
CfoNUS	Net Uniform Series
CfoPbk	Payback
CfoProf	Profitability Index
CfoTot	Total
choose	Choose
Clear	Memory
countX	Occurences
cos	Cosine
cosh	Hyperbolic Cosine
cot	Cotangent
csc	Cosecant
dDays	Difference Between Dates
degs	DMS to Degrees Conversion
degrees	Radians to Degrees Conversion
DepDBBV	Declining Balance Depreciation
DepDBDA	Declining Balance Depreciation
DepDBDV	Declining Balance Depreciation

DepDBSLBV	Declining Balance Crossover Depreciation
DepDBSLDA	Declining Balance Crossover Depreciation
DepDBSLDV	Declining Balance Crossover Depreciation
DepSLBV	Straight Line Depreciation
DepSLDA	Straight Line Depreciation
DepSLDV	Straight Line Depreciation
DepSOYDBV	Sum of the Year's Digits Depreciation
DepSOYDDA	Sum of the Year's Digits Depreciation
DepSOYDDV	Sum of the Year's Digits Depreciation
dms	Degrees to DMS Conversion
drop	Stack
dup	Stack
ex	Exponential
EE	Exponent
ENTER	Enter
floor	Floor
fMax	Maximum, Function
fMin	Minimum, Function
fnInt	Integral
fPart	Fractional Part
gcd	Greatest Common Denominator
getDate	Get Date in Decimal Format
getTime	Get Time in Decimal Format
HMS	Get Hours in HH.MMSS Format
HRS	Get Hours in Decimal Format
if	If
imag	Polar to Rectangular Conversion
IntEff	Effective Interest Rate
IntNorm	Nominal Interest Rate
invNorm	Inverse Cumulative Normal Distribution
iPart	Integer Part
last	Last
lcm	Least Common Multiple
ln	Natural Logarithm
log	Logarithm
makeDate	Make Date from Decimal Format

max	Maximum
mean	Mean
median	Median
min	Minimum
MEM	Memory
mod	Modulo Division
move	Stack
nCr	Combinations
nDeriv	First Derivative
nDeriv2	Second Derivative
nDist	Normal S Distribution
nPr	Permutations
prod	Product
radians	Degrees to Radians Conversion
rand	Random Number
randInt	Random Integer
RCL	Memory
Recall	Memory
real	Polar to Rectangular Conversion
rnd	Round
root	Root
rot	Stack
rrot	Stack
sec	Secant
seq	Sequence Evaluation
show	Show
sigma	Sigma Evaluation
sign	Sign
sin	Sine
sinh	Hyperbolic Sine
solving	Solving
sqrt	Square Root
stack	Stack
stdDev	Standard Deviation
stdDevP	Standard Deviation Population
STO	Memory

Store	Memory
sumX	Sum of X
sumX2	Sum of X squared
swap	Stack
tan	Tangent
tanh	Hyperbolic Tangent
toBool	Boolean, Convert To
today	Today
toFloat	Floating Point, Convert To
toInt	Integer, Convert To
TvmFV	Future Value
TvmI	Interest Rate
TvmN	Periods
TvmPmt	Payment
TvmPV	Present Value
USFV	Uniform Series Future Value
USPV	Uniform Series Present Value
variance	Variance
varianceP	Variance Population
weekDay	Day of the week
x	Multiplication
x!	Factorial
x^2	Square
xth root	Root
y^x	Power

5.4 A-B

This section covers functions beginning with the letters A through B.

5.4.1 Absolute Value

abs(valueA)

Returns absolute value of valueA.

Data Types: integer, floating point, table.

Category: number

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  abs(-15.6) : returns 15.6

- Order of Operations and Chain Input Modes

  15.6 +/- abs : returns 15.6

- RPN Input Mode

  15.6 ENT +/- abs : returns 15.6

5.4.2 Addition

valueA + valueB

Returns valueA plus valueB.

Data Types: boolean, integer, floating point, date, table.

Category: not applicable

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  5 + 4 : returns 9

  3.2 + 4.6 : returns 7.8

- Order of Operations and Chain Input Modes

  4 + 5 = : returns 9

  3.2 + 4.6 = : returns 7.8

- RPN Input Mode

  4 ENT 5 + : returns 9

  3.2 ENT 4.6 + : returns 7.8

5.4.3 Adjust Date

adjdate(date; days)

This function can only be used within a formula – the returned value cannot be viewed in a template. Returns a date type containing date plus or minus a number of days. date is a date type or a double containing the date in dd.mmyyyy format.

Category: date

Input Modes: solver

Examples:

adjDate(today(); -6) : returns 7/26/03 3:15 pm given today's date of 08/01/03 at 3:15 in the afternoon.

adjDate(today(); 2.5) : returns 8/4/03 03:15 am given today's date of 08/01/03 at 3:15 in the afternoon.

adjdate(date; days; months; years)

Same as above except adds months and years to the calculation.

Category: date

Input Modes: solver

Examples:

adjDate(01.082003; 6; 2; 2) : returns 10/7/05 12:00 am.

5.4.4 Adjust Time

adjtime(date; hours)

This function can only be used within a formula – the returned value cannot be viewed in a template. Returns a date type containing date plus or minus a number of hours. date is a date type, or a value in dd.mmyyyy format.

Category: date

Input Modes: solver

Examples:

adjTime(today(); -15) : returns 8/1/03 12:30 am given today's date of 08/01/03 at 3:30 in the afternoon.

adjTime(today(); 4.25) : returns 8/1/03 7:45 pm given today's date of 08/01/03 at 3:30 in the afternoon.

adjtime(date; hours; minutes; seconds)

Same as above except adds minutes and seconds to the calculation.

Category: date

Input Modes: solver

Examples:

adjTime(01.082003; 352; 6; 30) : returns 8/15/03 4:06 pm given a start time of 12:00 am on 08/01/2003

5.4.5 Amortization, End Balance

AmEndBal(period; PV; FV; I%; PMT; N)

Returns the ending principal balance of the given period. This function is only available if the finance library is installed.

- **period:** the period to calculate the ending principal balance.

- **PV:** present value. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **FV:** future value. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **I%:** yearly interest rate expressed as a percentage.
- **PMT:** periodic payment amount. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **N:** total number of periods.

Category: finance

Input Modes: solver

Examples:

AmEndBal(240; 150000; 0; 6; -899.325; 360) : returns 81,003.52

AmEndBal(period; PV; FV; I%; PMT; N; P/Y; C/Y; B; round)

Returns the ending principal balance of the given period. This function is only available if the finance library is installed. Same variables as above except:

- **P/Y:** payment periods per year. If not included, it is assumed to be 12.
- **C/Y:** interest compounding periods per year. If not included, it is assumed to be 12.
- **B:** payment timing (0 for end of period, 1 for beginning of period). If not included, it is assumed to be 0.
- **round:** decimal places to round the end balance to as it calculates. If not included, I is assumed to be 2.

Category: finance

Input Modes: solver

Examples:

AmEndBal(240; 150000; 0; 6; -899.325; 360; 12; 12; 0; 2) : returns 81,003.52

5.4.6 Amortization, Interest Paid

AmSumInt(first; last; PV; FV; I%; PMT; N)

Returns the summation of interest paid given a range of periods. This function is only available if the finance library is installed.

- **first:** calculate starting with this period.
- **last:** calculate ending with this period.
- **PV:** present value. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **FV:** future value. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **I%:** yearly interest rate expressed as a percentage.
- **PMT**: periodic payment amount. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **N:** total number of periods.

Category: finance

Input Modes: solver

Examples:

AmSumInt(1; 12; 150000; 0; 6; -899.325; 360) : returns –8,949.87

AmSumInt(first; last; PV; FV; I%; PMT; N; P/Y; C/Y; B; round)

Returns the summation of interest paid given a range of periods. This function is only available if the finance library is installed. Same variables as above except:

- **P/Y:** payment periods per year. If not included, it is assumed to be 12.

- **C/Y:** interest compounding periods per year. If not included, it is assumed to be 12.
- **B:** payment timing (0 for end of period, 1 for beginning of period). If not included, it is assumed to be 0.
- **round:** decimal places to round the end balance to as it calculates. If not included, I is assumed to be 2.

Category: finance

Input Modes: solver

Examples:

AmSumInt(1; 12; 150000; 0; 6; -899.325; 360; 12; 12; 0; 2) : returns –8,949.87

AmSumInt(first; last; PV; FV; I%; PMT; N; P/Y; C/Y; B; round; sign)

Returns the summation of interest paid given a range of periods. This function is only available if the finance library is installed. Same variables as above except:

- **sign:** set to false to leave the sign of the answer in the outputted format. Set to true to swap the sign.

Category: finance

Input Modes: solver

Examples:

AmSumInt(1; 12; 150000; 0; 6; -899.325; 360; 12; 12; 0; 2; true) : returns 8,949.87

AmSumInt(1; 12; 150000; 0; 6; -899.325; 360; 12; 12; 0; 2; false) : returns -8,949.87

5.4.7 Amortization, Principal Paid

AmSumPrn(first; last; PV; FV; I%; PMT; N)

Returns the summation of principal paid given a range of periods. This function is only available if the finance library is installed.

- **first:** calculate starting with this period.
- **last:** calculate ending with this period.
- **PV:** present value. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **FV:** future value. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **I%:** yearly interest rate expressed as a percentage.
- **PMT:** periodic payment amount. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **N:** total number of periods.

Category: finance

Input Modes: solver

Examples:

AmSumPrn(1; 12; 150000; 0; 6; -899.325; 360) : returns –1,842.09

AmSumPrn(first; last; PV; FV; I%; PMT; N; P/Y; C/Y; B; round)

Returns the summation of principal paid given a range of periods. This function is only available if the finance library is installed. Same variables as above except:

- **P/Y:** payment periods per year. If not included, it is assumed to be 12.
- **C/Y:** interest compounding periods per year. If not included, it is assumed to be 12.
- **B:** payment timing (0 for end of period, 1 for beginning of period). If not included, it is assumed to be 0.
- **round:** decimal places to round the end balance to as it calculates. If not included, I is assumed to be 2.

Category: finance

Input Modes: solver

Examples:

AmSumPrn(1; 12; 150000; 0; 6; -899.325; 360; 12; 12; 0; 2) : returns –1,842.09

AmSumPrn(first; last; PV; FV; I%; PMT; N; P/Y; C/Y; B; round; sign)

Returns the summation of principal paid given a range of periods. This function is only available if the finance library is installed. Same variables as above except:

- **sign:** set to false to leave the sign of the answer in the outputted format. Set to true to swap the sign.

Category: finance

Input Modes: solver

Examples:

AmSumPrn(1; 12; 150000; 0; 6; -899.325; 360; 12; 12; 0; 2; true) : returns 1,842.09

AmSumPrn(1; 12; 150000; 0; 6; -899.325; 360; 12; 12; 0; 2; false) : returns -1,842.09

5.4.8 And

valueA && valueB

Returns true if both valueA and valueB are true.

Data Types: boolean

Category: bool

Input Modes: solver

Examples:

0 && 5 : returns false

1 && 5 : returns true

(5 > 3) && (5 < 3) : returns false

(5 != 3) && (6 != 4) : returns true

5.4.9 Angle

Angle is the same as theta. See Rectangular to Polar Conversion for more information.

5.4.10 Arc-Cosine

acos(value)

Returns arc-cosine of value.

Data Types: integer, floating point, table.

Category: trig

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

acos(-.5) : returns 2.0944 when Trig Mode Preferences set to Radians

acos(-.5) : returns 120 when Trig Mode Preferences set to Degrees

- RPN, Order of Operations and Chain Input Modes

  .5 +/- acos : returns 2.0944 when Trig Mode Preferences set to Radians

  .5 +/- acos : returns 120 when Trig Mode Preferences set to Degrees

5.4.11 Arc-Sine

asin(value)

Returns arc-sine of value.

Data Types: integer, floating point, table.

Category: trig

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  asin(.5) : returns 0.5236 when Trig Mode Preferences set to Radians

  asin(.5) : returns 30 when Trig Mode Preferences set to Degrees

- RPN, Order of Operations and Chain Input Modes

  .5 asin : returns 0.5236 when Trig Mode Preferences set to Radians

  .5 asin : returns 30 when Trig Mode Preferences set to Degrees

5.4.12 Arc-Tangent

atan(value)

Returns arc-tangent of value.

Data Types: integer, floating point, table

Category: trig

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  atan(1) : returns 0.7854 when Trig Mode Preferences set to Radians

  atan(1) : returns 45 when Trig Mode Preferences set to Degrees

- RPN, Order of Operations and Chain Input Modes

  1 atan : returns 0.7854 when Trig Mode Preferences set to Radians

  1 atan : returns 45 when Trig Mode Preferences set to Degrees

5.4.13 ARM Amortization, End Balance

ArmEndBal(period; PV; FV; startRate; startPmt; N; rateDelta; interestOnlyPeriods; initialRatePeriods; rateAdjInterval; firstRateAdjUpLimit; firstRateAdjDownLimit; periodRateAdjUpLimit; periodRateAdjDownLimit; annualRateAdjUpLimit; annualRateAdjDownLimit; totalRateAdjUpLimit; totalRateAdjDownLimit)

Returns the ending principal balance of the given period for an adjustable rate loan. This function is only available if the finance library is installed.

- **period:** the period to calculate the ending principal balance.
- **PV:** present value. Positive values mean a cash inflow while negative numbers mean a cash outflow. See
- **FV:** future value. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **startRate:** initial interest rate used until first adjustment is made expressed as a percentage.
- **startPmt:** initial payment used until first adjustment is made. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **N:** total number of periods.
- **rateDelta:** simulated interest rate change per year expressed as a percentage.
- **interestOnlyPeriods:** number of interest-only periods before beginning amortization.
- **initialRatePeriods:** number of periods at initial rate before first rate adjustment.
- **rateAdjInterval:** number of periods between subsequent rate adjustments. 12 means adjustments will be made on period 13, 25, 37, etc.
- **firstRateAdjUpLimit:** upper limit for first rate adjustment expressed as a positive percentage.
- **firstRateAdjDownLimit:** lower limit for first rate adjustment expressed as a positive percentage.
- **periodRateAdjUpLimit:** upper limit for rate adjustment per subsequent period expressed as a positive percentage.
- **periodRateAdjDownLimit:** lower limit for rate adjustment per subsequent period expressed as a positive percentage.
- **annualRateAdjUpLimit:** upper limit for total rate adjustments per loan year expressed as a positive percentage.
- **annualRateAdjDownLimit:** lower limit for total rate adjustments per loan year expressed as a positive percentage.
- **totalRateAdjUpLimit:** upper limit for total rate adjustments over life of loan expressed as a positive percentage.
- **totalRateAdjDownLimit:** lower limit for total rate adjustments over life of loan in expressed as a positive percentage.

Category: finance

Input Modes: solver

ArmEndBal(period; PV; FV; startRate; startPmt; N; P/Y; C/Y; round; rateDelta; interestOnlyPeriods; initialRatePeriods; rateAdjInterval; firstRateAdjUpLimit; firstRateAdjDownLimit; periodRateAdjUpLimit; periodRateAdjDownLimit; annualRateAdjUpLimit; annualRateAdjDownLimit; totalRateAdjUpLimit; totalRateAdjDownLimit)

Returns the ending principal balance of the given period for an adjustable rate loan. This function is only available if the finance library is installed. Same variables as above except:

- **P/Y:** payment periods per year. If not included, it is assumed to be 12.
- **C/Y:** interest compounding periods per year. If not included, it is assumed to be 12.
- **round:** decimal places to round the end balance to as it calculates. If not included, it is assumed to be 2.

Category: finance

Input Modes: solver

ArmEndBal(period; PV; FV; startRate; startPmt; N;
rateDelta; interestOnlyPeriods; initialRatePeriods; rateAdjInterval;
firstRateAdjUpLimit; firstRateAdjDownLimit;
periodRateAdjUpLimit; periodRateAdjDownLimit;
annualRateAdjUpLimit; annualRateAdjDownLimit;
totalRateAdjUpLimit; totalRateAdjDownLimit;
initialPmtPeriods; pmtAdjInterval; periodPmtAdjUpLimit;
annualPmtAdjUpLimit; totalPmtAdjUpLimit;
negativeAmort; negativeAmortLimit)

Returns the ending principal balance of the given period for an adjustable rate loan. This function is only available if the finance library is installed. This function excludes P/Y, C/Y, and round and adds the following variables:

- **initialPmtPeriods:** number of periods at initial payment value before first payment adjustment. If not included, it is assumed to be 0.
- **pmtAdjInterval:** number of periods between subsequent payment adjustments. 12 means adjustments will be made on period 13, 25, 37, etc. If not included, it is assumed to be 1.
- **periodPmtAdjUpLimit:** upper limit for payment adjustment per period expressed as a percentage.
- **annualPmtAdjUpLimit:** upper limit for total payment adjustments per loan year expressed as a percentage.
- **totalPmtAdjUpLimit:** upper limit for total payment adjustments over life of loan expressed as a percentage.
- **negativeAmort:** true means if payment is capped at too low of a value to cover the interest for the period, unpaid interest is added to balance and loan amount grows. If false and payments don't cover interest, excess interest is forfeited by lender and balance stays steady. If not included, it is assumed to be false.
- **negativeAmortLimit:** negative amortization limit expressed as a percentage of the original loan amount. 25 means that if negative amortization would cause the loan balance to become >= 125% of the original loan amount, then all payment caps are removed and the loan switches to fully amortizing for the remaining periods.

Category: finance

Input Modes: solver

ArmEndBal(period; PV; FV; startRate; startPmt; N; P/Y; C/Y; round;
rateDelta; interestOnlyPeriods; initialRatePeriods; rateAdjInterval;
firstRateAdjUpLimit; firstRateAdjDownLimit;
periodRateAdjUpLimit; periodRateAdjDownLimit;
annualRateAdjUpLimit; annualRateAdjDownLimit;
totalRateAdjUpLimit; totalRateAdjDownLimit;
initialPmtPeriods; pmtAdjInterval; periodPmtAdjUpLimit;
annualPmtAdjUpLimit; totalPmtAdjUpLimit;
negativeAmort; negativeAmortLimit)

Returns the ending principal balance of the given period for an adjustable rate loan. This function is only available if the finance library is installed. This function includes all variables as defined above.

5.4.14 ARM Amortization, Interest Paid

ArmSumInt (first; last; PV; FV; startRate; startPmt; N;
rateDelta; interestOnlyPeriods; initialRatePeriods; rateAdjInterval;
firstRateAdjUpLimit; firstRateAdjDownLimit;
periodRateAdjUpLimit; periodRateAdjDownLimit;
annualRateAdjUpLimit; annualRateAdjDownLimit;
totalRateAdjUpLimit; totalRateAdjDownLimit)

Returns the sum of the interest for the given periods for an adjustable rate loan. This function is only available if the finance library is installed.

- **first:** The start period to calculate the interest sum.
- **last:** The start period to calculate the interest sum.
- **PV:** present value. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **FV:** future value. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **startRate:** initial interest rate used until first adjustment is made expressed as a percentage.
- **startPmt:** initial payment used until first adjustment is made. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **N:** total number of periods.
- **rateDelta:** simulated interest rate change per year expressed as a percentage.
- **interestOnlyPeriods:** number of interest-only periods before beginning amortization.
- **initialRatePeriods:** number of periods at initial rate before first rate adjustment.
- **rateAdjInterval:** number of periods between subsequent rate adjustments. 12 means adjustments will be made on period 13, 25, 37, etc.
- **firstRateAdjUpLimit:** upper limit for first rate adjustment expressed as a positive percentage.
- **firstRateAdjDownLimit:** lower limit for first rate adjustment expressed as a positive percentage.
- **periodRateAdjUpLimit:** upper limit for rate adjustment per subsequent period expressed as a positive percentage.
- **periodRateAdjDownLimit:** lower limit for rate adjustment per subsequent period expressed as a positive percentage.
- **annualRateAdjUpLimit:** upper limit for total rate adjustments per loan year expressed as a positive percentage.
- **annualRateAdjDownLimit:** lower limit for total rate adjustments per loan year expressed as a positive percentage.
- **totalRateAdjUpLimit:** upper limit for total rate adjustments over life of loan expressed as a positive percentage.
- **totalRateAdjDownLimit:** lower limit for total rate adjustments over life of loan in expressed as a positive percentage.

Category: finance

Input Modes: solver

ArmSumInt (first; last; PV; FV; startRate; startPmt; N; P/Y; C/Y; round; flipSign; rateDelta; interestOnlyPeriods; initialRatePeriods; rateAdjInterval; firstRateAdjUpLimit; firstRateAdjDownLimit; periodRateAdjUpLimit; periodRateAdjDownLimit; annualRateAdjUpLimit; annualRateAdjDownLimit; totalRateAdjUpLimit; totalRateAdjDownLimit)

Returns the sum of the interest for the given periods for an adjustable rate loan. This function is only available if the finance library is installed. Same variables as above except:

- **P/Y:** payment periods per year. If not included, it is assumed to be 12.
- **C/Y:** interest compounding periods per year. If not included, it is assumed to be 12.
- **round:** decimal places to round the end balance to as it calculates. If not included, it is assumed to be 2.
- **flipSign:** If not included, it is assumed to be false.

Category: finance

Input Modes: solver

ArmSumInt (first; last; PV; FV; startRate; startPmt; N; rateDelta; interestOnlyPeriods; initialRatePeriods; rateAdjInterval; firstRateAdjUpLimit; firstRateAdjDownLimit; periodRateAdjUpLimit; periodRateAdjDownLimit; annualRateAdjUpLimit; annualRateAdjDownLimit;

**totalRateAdjUpLimit; totalRateAdjDownLimit;
initialPmtPeriods; pmtAdjInterval; periodPmtAdjUpLimit;
annualPmtAdjUpLimit; totalPmtAdjUpLimit;
negativeAmort; negativeAmortLimit)**

Returns the sum of the interest for the given periods for an adjustable rate loan. This function is only available if the finance library is installed. This function excludes P/Y, C/Y, round and flipsign and adds the following variables:

- **initialPmtPeriods:** number of periods at initial payment value before first payment adjustment. If not included, it is assumed to be 0.
- **pmtAdjInterval:** number of periods between subsequent payment adjustments. 12 means adjustments will be made on period 13, 25, 37, etc. If not included, it is assumed to be 1.
- **periodPmtAdjUpLimit:** upper limit for payment adjustment per period expressed as a percentage.
- **annualPmtAdjUpLimit:** upper limit for total payment adjustments per loan year expressed as a percentage.
- **totalPmtAdjUpLimit:** upper limit for total payment adjustments over life of loan expressed as a percentage.
- **negativeAmort:** true means if payment is capped at too low of a value to cover the interest for the period, unpaid interest is added to balance and loan amount grows. If false and payments don't cover interest, excess interest is forfeited by lender and balance stays steady. If not included, it is assumed to be false.
- **negativeAmortLimit:** negative amortization limit expressed as a percentage of the original loan amount. 25 means that if negative amortization would cause the loan balance to become >= 125% of the original loan amount, then all payment caps are removed and the loan switches to fully amortizing for the remaining periods.

Category: finance

Input Modes: solver

**ArmSumInt (first; last; PV; FV; startRate; startPmt; N; P/Y; C/Y; round; flipSign;
rateDelta; interestOnlyPeriods; initialRatePeriods; rateAdjInterval;
firstRateAdjUpLimit; firstRateAdjDownLimit;
periodRateAdjUpLimit; periodRateAdjDownLimit;
annualRateAdjUpLimit; annualRateAdjDownLimit;
totalRateAdjUpLimit; totalRateAdjDownLimit;
initialPmtPeriods; pmtAdjInterval; periodPmtAdjUpLimit;
annualPmtAdjUpLimit; totalPmtAdjUpLimit;
negativeAmort; negativeAmortLimit)**

Returns the sum of the interest for the given periods for an adjustable rate loan. This function is only available if the finance library is installed. This function includes all variables as defined above.

5.4.15 ARM Amortization, Principal Paid

**ArmSumPrn(first; last; PV; FV; startRate; startPmt; N;
rateDelta; interestOnlyPeriods; initialRatePeriods; rateAdjInterval;
firstRateAdjUpLimit; firstRateAdjDownLimit;
periodRateAdjUpLimit; periodRateAdjDownLimit;
annualRateAdjUpLimit; annualRateAdjDownLimit;
totalRateAdjUpLimit; totalRateAdjDownLimit)**

Returns the total principal paid for the given periods for an adjustable rate loan. This function is only available if the finance library is installed.

- **first:** The start period to calculate the interest sum.
- **last:** The start period to calculate the interest sum.
- **PV:** present value. Positive values mean a cash inflow while negative numbers mean a cash outflow.

- **FV:** future value. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **startRate:** initial interest rate used until first adjustment is made expressed as a percentage.
- **startPmt:** initial payment used until first adjustment is made. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **N:** total number of periods.
- **rateDelta:** simulated interest rate change per year expressed as a percentage.
- **interestOnlyPeriods:** number of interest-only periods before beginning amortization.
- **initialRatePeriods:** number of periods at initial rate before first rate adjustment.
- **rateAdjInterval:** number of periods between subsequent rate adjustments. 12 means adjustments will be made on period 13, 25, 37, etc.
- **firstRateAdjUpLimit:** upper limit for first rate adjustment expressed as a percentage.
- **firstRateAdjDownLimit:** lower limit for first rate adjustment expressed as a percentage.
- **periodRateAdjUpLimit:** upper limit for rate adjustment per subsequent period expressed as a positive percentage.
- **periodRateAdjDownLimit:** lower limit for rate adjustment per subsequent period expressed as a positive percentage.
- **annualRateAdjUpLimit:** upper limit for total rate adjustments per loan year expressed as a positive percentage.
- **annualRateAdjDownLimit:** lower limit for total rate adjustments per loan year expressed as a positive percentage.
- **totalRateAdjUpLimit:** upper limit for total rate adjustments over life of loan expressed as a positive percentage.
- **totalRateAdjDownLimit:** lower limit for total rate adjustments over life of loan in expressed as a positive percentage.

Category: finance

Input Modes: solver

ArmSumPrn(first; last; PV; FV; startRate; startPmt; N; P/Y; C/Y; round; flipSign; rateDelta; interestOnlyPeriods; initialRatePeriods; rateAdjInterval; firstRateAdjUpLimit; firstRateAdjDownLimit; periodRateAdjUpLimit; periodRateAdjDownLimit; annualRateAdjUpLimit; annualRateAdjDownLimit; totalRateAdjUpLimit; totalRateAdjDownLimit)

Returns the total principal paid for the given periods for an adjustable rate loan. This function is only available if the finance library is installed. Same variables as above except:

- **P/Y:** payment periods per year. If not included, it is assumed to be 12.
- **C/Y:** interest compounding periods per year. If not included, it is assumed to be 12.
- **round:** decimal places to round the end balance to as it calculates. If not included, it is assumed to be 2.
- **flipSign:** If not included, it is assumed to be false.

Category: finance

Input Modes: solver

ArmSumPrn(first; last; PV; FV; startRate; startPmt; N; rateDelta; interestOnlyPeriods; initialRatePeriods; rateAdjInterval; firstRateAdjUpLimit; firstRateAdjDownLimit; periodRateAdjUpLimit; periodRateAdjDownLimit; annualRateAdjUpLimit; annualRateAdjDownLimit; totalRateAdjUpLimit; totalRateAdjDownLimit; initialPmtPeriods; pmtAdjInterval; periodPmtAdjUpLimit; annualPmtAdjUpLimit; totalPmtAdjUpLimit; negativeAmort; negativeAmortLimit)

Returns the total principal paid for the given periods for an adjustable rate loan. This function is only available if the

finance library is installed. This function excludes P/Y, C/Y, round and flipsign and adds the following variables:

- **initialPmtPeriods:** number of periods at initial payment value before first payment adjustment. If not included, it is assumed to be 0.
- **pmtAdjInterval:** number of periods between subsequent payment adjustments. 12 means adjustments will be made on period 13, 25, 37, etc. If not included, it is assumed to be 1.
- **periodPmtAdjUpLimit:** upper limit for payment adjustment per period expressed as a percentage.
- **annualPmtAdjUpLimit:** upper limit for total payment adjustments per loan year expressed as a percentage.
- **totalPmtAdjUpLimit:** upper limit for total payment adjustments over life of loan expressed as a percentage.
- **negativeAmort:** true means if payment is capped at too low of a value to cover the interest for the period, unpaid interest is added to balance and loan amount grows. If false and payments don't cover interest, excess interest is forfeited by lender and balance stays steady. If not included, it is assumed to be false.
- **negativeAmortLimit:** negative amortization limit expressed as a percentage of the original loan amount. 25 means that if negative amortization would cause the loan balance to become >= 125% of the original loan amount, then all payment caps are removed and the loan switches to fully amortizing for the remaining periods.

Category: finance

Input Modes: solver

ArmSumPrn(first; last; PV; FV; startRate; startPmt; N; P/Y; C/Y; round; flipSign; rateDelta; interestOnlyPeriods; initialRatePeriods; rateAdjInterval; firstRateAdjUpLimit; firstRateAdjDownLimit; periodRateAdjUpLimit; periodRateAdjDownLimit; annualRateAdjUpLimit; annualRateAdjDownLimit; totalRateAdjUpLimit; totalRateAdjDownLimit; initialPmtPeriods; pmtAdjInterval; periodPmtAdjUpLimit; annualPmtAdjUpLimit; totalPmtAdjUpLimit; negativeAmort; negativeAmortLimit)

Returns the total principal paid for the given periods for an adjustable rate loan. This function is only available if the finance library is installed. This function includes all variables as defined above.

5.4.16 Backspace

←

Moves backwards one space, deleting the item before the cursor (in algebraic and RPN input mode) or the last entered number in order of operations and chain input modes.

Category: not applicable

Input Modes: solver, RPN, order of operations, chain

Examples:

345.67 ← : shows 345.6

5.4.17 Bond Accrued Interest

BondA(SD; MD; C/Y; CR)

Returns the accumulated interest of a bond. This function is only available if the finance library is installed.

- **SD:** settlement date entered in dd.mmyyyy format
- **MD:** maturity date entered in dd.mmyyyy format
- **CY:** number of compounding interest periods per year

- **CR:** coupon rate expressed as a percentage

Category: finance

Input Modes: solver

Examples:

BondA(1.012003; 15.122010; 4; 6) : returns 0.283

5.4.18 Bond Price

BondP(SD; MD; C/Y; CR; RV; Y)

Returns the price of a bond. This function is only available if the finance library is installed.

- **SD:** settlement date entered in dd.mmyyyy format
- **MD:** maturity date entered in dd.mmyyyy format
- **CY:** number of compounding interest periods per year
- **CR:** coupon rate expressed as a percentage
- **RV:** residual value
- **Y:** annual yield expressed as a percentage

Category: finance

Input Modes: solver

Examples:

BondP(1.012003; 31.122010; 4; 6; 100; 5.25) : returns 104.87

5.4.19 Bond Yield

BondY(SD; MD; C/Y; CR; RV; P)

Returns the yield of a bond expressed as a percentage. This function is only available if the finance library is installed.

- **SD:** settlement date entered in dd.mmyyyy format
- **MD:** maturity date entered in dd.mmyyyy format
- **CY:** number of compounding interest periods per year
- **CR:** coupon rate expressed as a percentage
- **RV:** residual value
- **P:** price

Category: finance

Input Modes: solver
Examples:

BondY(1.012003; 31.122010; 4; 6; 100; 104) : returns 5.38

5.4.20 Boolean, Convert To

tobool(value)

Returns a boolean by converting value to a boolean value.

Data Types: boolean, integer, floating point

Category: number

Input Modes: solver
Examples:

tobool(0) : returns false (0)

tobool(5 - 2) : returns true (1)

tobool(5 / 2) : returns true (1)

tobool(0) + tobool(5 - 2) + tobool(5 / 2) : returns 2

5.4.21 Braces { }

{valueA; valueB;….} for a list

{ {valueA1; valueA2; ...}; {valueB1; valueB2;...} } for a table

Use braces to create a list or table. When creating a table, inside braces group rows.

Category: stats

Input Modes: solver

Examples:

{1; 2; 3} : returns list

1
2
3

{ {1}; {2}; {3} } : returns list

1	2	3

{ {1; 2}; {3; 4} } : returns table

1	2
3	4

5.5 C

This section covers functions beginning with the letter C.

5.5.1 Ceiling

ceil(value)

Returns the smallest integer greater than or equal to value.

Data Types: integer, floating point, table

Category: number

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  ceil(4.5) : returns 5

  ceil(-4.5) : returns -4

  ceil({ {-15.2; 22}; {8.25; -89.9} }) : returns { {-15; 22}; {9; -89} }

- Order of Operations and Chain Input Modes

  4.5 ceil : returns 5

  4.5 +/- ceil : returns -4

- RPN Input Mode

  4.5 ceil : returns 5

  4.5 ENT +/- ceil : returns -4

5.5.2 Choose

choose(index; expression1; ...expressionN)

Uses index to pick one of the expressions following index then returns the value of the picked expression. index can be a number or an expression (as long as the expression can be evaluated), where 1 <= index <= number of expressions supplied.

Category: bool

Input Modes: solver

Examples:

choose(3; 2; (3*5); 4; 5; "six"; 7) : returns 4

choose(2; 2; (3*5); 4; 5; "six"; 7) : returns 15

choose((15/3); 2; (3*5); 4; 5; "six"; 7) : returns "six"

choose(A; 2; (3*5); 4; 5; "six"; 7) : returns 2 where A is a variable equal to 1

5.5.3 Clear

CE/C

CE/C performs two different clear functions. If a number is currently being entered, selecting CE/C clears the current entry. If a mathematics symbol, equals or enter has been selected, clear clears the current calculation.

Category: not applicable

Input Modes: solver, RPN, order of operations, chain

Clear

See Memory for additional information.

5.5.4 Combinations

nCr(n; r)

Returns the number of combinations of n taken r at a time. n, r must be integer values where r <= n, 0 <= n, r <= 170. Returned values correspond to n!/(r!(n-r)!)

Data Types: integer, floating point, table

Category: prob

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

nCr(8; 3) : returns 56

nCr({10; 11; 12}; 2) : returns {45; 55; 66}

- Order of Operations and Chain Input Modes

  8 nCr 3 = : returns 56

- RPN Input Mode

  8 ENT 3 nCr : returns 56

5.5.5 Cosecant

csc(value)

Returns cosecant of value.

Data Types: integer, floating point, table

Category: trig

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  csc(0.5236) : returns 2 when Trig Mode Preferences set to Radians

  csc(30) : returns 2 when Trig Mode Preferences set to Degrees

- Order of Operations, Chain and RPN Input Modes

  0.5236 csc : returns 2 when Trig Mode Preferences set to Radians

  30 csc : returns 2 when Trig Mode Preferences set to Degrees

5.5.6 Cosine

cos(value)

Returns cosine of value.

Data Types: integer, floating point, table

Category: trig

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  cos(30) : returns 0.1543 when Trig Mode Preferences set to Radians

  cos(30) : returns 0.866 when Trig Mode Preferences set to Degrees

- RPN, Order of Operations and Chain Input Modes

  30 cos : returns 0.1543 when Trig Mode Preferences set to Radians

  30 cos : returns 0.866 when Trig Mode Preferences set to Degrees

5.5.7 Cotangent

cot(value)

Returns cotangent of value.

Data Types: integer, floating point, table

Category: trig

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  cot(0.5236) : returns 1.7321 when Trig Mode Preferences set to Radians

  cot(30) : returns 1.7321 when Trig Mode Preferences set to Degrees

- Order of Operations, Chain and RPN Input Modes

  0.5236 cot : returns 1.7321 when Trig Mode Preferences set to Radians

  30 cot : returns 1.7321 when Trig Mode Preferences set to Degrees

5.5.8 Count

CfoCount(CFAmntList)

Returns the total number of periods in the given cash flow excluding the initial cash flow. This function is only available if the finance library is installed.

- **CFAmntList:** list containing cash flow amounts where the first element is the initial cash flow.
- **CFFreqList:** list in which each element specifies the frequency of occurrence for a consecutive cash flow amount in CFList.

Category: finance

Input Modes: solver

Examples:

CfoCount({-5000; 4000; 3000; 3000}) : returns 3

CfoCount(CFAmntList; CFFreqList)

same as above

Category: finance

Input Modes: solver

Examples:

CfoCount({-5000; 4000; 3000}; {1; 1; 2}) : returns 3

CfoCount({-5000; 4000; 3000}; {2; 1; 2}) : returns 4

See also Occurrences.

5.5.9 Cubed Root

cbrt(value)

Returns the cube root of value.

Data Types: integer, floating point, table

Category: math

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  cbrt(5) : returns 1.71

  cbrt(87.6) : returns 4.4412

- Order of Operations, Chain and RPN Input Modes

  5 $\sqrt[3]{x}$: returns 1.71

  87.6 $\sqrt[3]{x}$: returns 4.4412

5.5.10 Cumulative Standard Normal Distribution

NormSDist(upperlimit)

Returns the area under the standard normal distribution curve (mean = 0, standard deviation = 1) bounded by an upper limit.

Data Types: integer, floating point, table

Category: distr

Input Modes: solver

Examples:

- Solver Input Mode

  nDist(0.5) : returns 0.6915

  NormSDist(0.5) : returns 0.6915

5.6 D-F

This section covers functions beginning with the letters D through F.

5.6.1 Day of Week

wkday(date)

Returns a number representing the day of the week (1 = Sunday, 7 = Saturday). Date must be a date type or a value in dd.mmyyyy format.

Data Types: floating point, date

Category: date

Input Modes: solver

Examples:

wkday(today()) : returns today's weekday number (e.g, 6 for Friday)

wkday(20.071969) : returns 1 (Sunday)

choose(wkday(15.082003);"Sun";"Mon";"Tue";"Wed";Thu";"Fri";"Sat") : returns "Fri"

5.6.2 Decimal Separator

. (decimal point)
, (decimal comma)

Separates the integer and fractional portion of the number. How numbers appear is set in the system preferences. See the device users manual for additional information.

Category: not applicable

Input Modes: solver, RPN, order of operations, chain

5.6.3 Declining Balance Crossover Depreciation

DepDBSLBV (C; S; L; M; Y; R) (Book Value)
DepDBSLDA (C; S; L; M; Y; R) (Depreciation Amount)
DepDBSLDV (C; S; L; M; Y; R) (Depreciation Value)

Book value returns the book value (depreciable value + salvage value) for the asset at the end of the given year. Depreciation amount returns the amount that the asset depreciated during the given year. Depreciation value returns the remaining total depreciable value for the asset at the end of the given year. All three are calculated using the declining balance crossover to straight line method of depreciation. This function is only available if the finance library is installed.

- **C:** cost of the depreciable asset
- **S:** salvage value of the depreciable asset
- **L:** life in years of the depreciable asset
- **M:** first month to begin depreciating (1 is January, 12 is December)
- **Y:** year to calculate
- **R:** depreciation rate expressed as a percentage

Category: finance

Input Modes: algebraic, RPN

Examples:

- Algebraic Input Mode

  DepDBSLBV (150000; 20000; 20; 6; 3; 200) : returns 114,412.50

  DepDBSLDA (150000; 20000; 20; 6; 3; 200) : returns 12,712.50

  DepDBSLDV (150000; 20000; 20; 6; 3; 200) : returns 94,412.50

- RPN Input Mode (HP48 Enter Mode Preference setting)

  150000 ENT 20000 ENT 20 ENT 6 ENT 3 ENT 200 DepDBSLBV : returns 114,412.50

  150000 ENT 20000 ENT 20 ENT 6 ENT 3 ENT 200 DepDBSLDA : returns 12,712.50

  150000 ENT 20000 ENT 20 ENT 6 ENT 3 ENT 200 DepDBSLDV : returns 94,412.50

5.6.4 Declining Balance Depreciation

DepDBBV (C; S; L; M; Y; R) (Book Value)
DepDBDA (C; S; L; M; Y; R) (Depreciation Amount)
DepDBDV (C; S; L; M; Y; R) (Depreciation Value)

Book value returns the book value (depreciable value + salvage value) for the asset at the end of the given year. Depreciation amount returns the amount that the asset depreciated during the given year. Depreciation value returns the remaining total depreciable value for the asset at the end of the given year. All three are calculated using the declining balance method of depreciation. This function is only available if the finance library is installed.

- **C:** cost of the depreciable asset
- **S:** salvage value of the depreciable asset
- **L:** life in years of the depreciable asset
- **M:** first month to begin depreciating (1 is January, 12 is December)
- **Y:** year to calculate
- **R:** depreciation rate expressed as a percentage

Category: finance

Input Modes: solver

Examples:

DepDBBV (150000; 20000; 20; 6; 3; 200) : returns 114,412.50

DepDBDA (150000; 20000; 20; 6; 3; 200) : returns 12,712.50

DepDBDV (150000; 20000; 20; 6; 3; 200) : returns 94,412.50

5.6.5 Degrees to DMS Conversion

dms(value)

Returns equivalent in dd.mmss (degrees, minutes, seconds) of value degrees.

Data Types: integer, floating point, table

Category: number

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  dms(90.50) : returns 90.30

  dms({44.6333; 121.1333}) : returns {44.376; 121.076}

- Order of Operations and Chain Input Modes

  90.50 dms : returns 90.30

- RPN Input Mode

  90.50 dms : returns 90.30

5.6.6 Degrees to Radians Conversion

radians(value)

Returns equivalent in radians of value degrees.

Data Types: integer, floating point, table

Category: trig

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  radians(360) : returns 6.2832

- RPN, Order of Operations and Chain Input Modes

  360 radians : returns 6.2832

5.6.7 Derivative

nDeriv("expression"; "variable"; value)

Returns an approximate numerical derivative of expression with respect to variable at value. This function uses the "tolerance" constant.

- **expression:** the expression to analyze. Must be in quotations.
- **variable:** the variable within the expression. Must be in quotations.
- **value:** point to evaluate.

Category: calc

Input Modes: algebraic

Examples:

nDeriv("x^2"; "x"; 3) : returns 6

nDeriv("expression"; "variable"; value; e)

Same as above except with specified tolerance e.

- **e:** tolerance. If it is not included, it defaults to 10^{-4}.

Category: calc

Input Modes: algebraic, RPN

Examples:

- Algebraic Input Mode

  nDeriv("x^2"; "x"; 3; 0.001) : returns 6

- RPN Input Mode

  "x^2" ENT "x" ENT 3 ENT 0.001 nDeriv : returns 6

5.6.8 Derivative, Second

nDeriv2("expression"; "variable"; value)

Returns an approximate second numerical derivative of expression with respect to variable at value. This function uses the "tolerance" constant.

- **expression:** the expression to analyze. Must be in quotations.
- **variable:** the variable within the expression. Must be in quotations.

- **value:** point to evaluate.

Category: calc

Input Modes: algebraic

Examples:

nDeriv2("x^3"; "x"; 3) : returns 18

nDeriv2("expression"; "variable"; value; e)

Same as above except with specified tolerance e.

- **e:** tolerance. If it is not included, it defaults to 10^{-4}.

Category: calc

Input Modes: algebraic, RPN

Examples:

- Algebraic Input Mode

  nDeriv2("x^2"; "x"; 3; 0.001) : returns 18

- RPN Input Mode

  "x^2" ENT "x" ENT 3 ENT 0.001 nDeriv : returns 6

5.6.9 Difference Between Dates

ddays(date1; date2)

date1, date2 must be a date type or a value in dd.mmyyyy format. Returns a value representing the number of days between two dates.

Data Types: floating point, date

Category: date

Input Modes: solver

Examples:

ddays(01.082003;20.071969) : returns -12,430 in days.

5.6.10 Division

valueA / valueB

Returns valueA divided by valueB.

Data Types: integer, floating point, table

Category: not applicable

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  88 / 44 : returns 2

  18.6 / 4.5 : returns 4.1333

- Order of Operations and Chain Input Modes

  88 / 44 = : returns 2

  18.6 / 4.5 = : returns 4.1333

- RPN Input Mode

  88 ENT 44 / : returns 2

  18.6 ENT 4.5 / : returns 4.1333

5.6.11 DMS to Degrees Conversion

degs(value)

Returns equivalent in degrees of value dd.mmss (ie degrees, minutes, seconds format).

Data Types: integer, floating point, table

Category: number

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  degs(90.30) : returns 90.50

  degs({44.38; 121.08}) : returns {44.6333; 121.1333}

- Order of Operations and Chain Input Modes

  90.30 degs : returns 90.50

- RPN Input Mode

  90.30 degs : returns 90.50

5.6.12 Effective Interest Rate

EffNom(rate; compoundingperiods)

Returns the effective interest rate. This function is only available if the finance library is installed.

- **rate:** nominal annual interest rate. Must be an integer or floating point number.
- **compoundingperiods:** number of interest compounding periods per year. Must be an integer or floating point number greater than or equal to 0. 0 denotes continuous compounding.

Category: solver

Input Modes: solver

Examples:

IntEff(7; 360) : returns 7.250

IntEff(7; 0) : returns 7.251

5.6.13 Enter

ENT

Used to complete calculations in RPN input mode. See the Using the Calculator : Input Modes section for more

information.

5.6.14 Equals

valueA == valueB

Returns true if valueA is equal to value valueB. Note that this operator is entered using two consecutive "=" characters, not a single "=" character.

Data Types: boolean, integer, floating point, date

Category: bool

Input Modes: solver

Examples:

5 == 5 : returns true

5 == 1 - 5 : returns false

=

Used to complete calculations in order of operations and chain input modes. See the Using the Calculator : Input Modes section for more information.

5.6.15 Exclusive Or

valueA ## valueB

Returns true if valueA is true or valueB is true but both are not true.

Data Types: boolean

Category: bool

Input Modes: solver

Examples:

- Solver Input Mode

  0 ## 1 : returns true

  1 ## 1 : returns false

  0 ## 0 : returns false

  (5>3) ## (5<3) : returns true

  (5>3) ## (5>4) : returns false

5.6.16 Exponent

value E exponent

Used to make value times 10 raised to exponent where exponent is an integer (whole number). value*10exponent must lie between 1E-308 and 1E308 inclusive.

Data Types: integer, floating point

Category: number

Input Modes: solver, RPN, order of operations, chain

Examples:

5E3 : equivalent to 5000

314 E +/- 2 = : equivalent to 3.14

5.6.17 Exponential

exp(value)

Returns e raised to the value power.

Data Types: integer, floating point, table

Category: math

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  e ^ 4 : returns 54.5982

  e ^ 6.85 : returns 943.8809

- Order of Operations and Chain Input Modes

  4 e^ : returns 54.5982

  6.85 e^ : returns 943.8809

- RPN Input Mode

  4 e^ : returns 54.5982

  6.85 e^ : returns 943.8809

5.6.18 Factorial

fact(value)

Returns factorial of value, where -169 < value <= 170.

Data Types: integer (positive numbers only), floating point, table

Category: prob

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  fact(6) : returns 720

- Order of Operations, Chain and RPN Input Modes

  6 ! : returns 720

5.6.19 Floating Point, Convert To

tofloat(value)

Returns a floating point number by converting value.

Data Types: boolean, integer, floating point

Category: number

Input Modes: solver

Examples:

toFloat(5) : returns 5.0

5.6.20 Floor

floor(value)

Returns the largest integer less than or equal to value.

Data Types: integer, floating point, complex, table

Category: number

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  floor(4.5) : returns 4

  floor(-4.5) : returns -5

  floor({ {-15.2; 22}; {8.25; -89.9} }) : returns { {-16; 22}; {8; -90} }

- Order of Operations and Chain Input Modes

  4.5 floor : returns 4

  4.5 +/- floor : returns -5

- RPN Input Mode

  4.5 floor : returns 4

  4.5 ENT +/- floor : returns -5

5.6.21 Fractional Part

fpart(value)

Returns fractional part of value.

Data Types: integer, floating point, complex, table, matrix

Category: number

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  fpart(5) : returns 0.0

  fpart(5.25) : returns .25

  fpart({ {1.56; 2.89}; {3.45; 4.73} }) : returns { {0.56; 0.89}; {0.45; 0.73} }

- Order of Operations and Chain Input Modes

  5 fPart : returns 0.0

5.25 fPart : returns 0.25

- RPN Input Mode

  5 fPart : returns 0.0

  5.25 fPart : returns 0.25

5.6.22 Future Value

tvmfv(N; I%; PV; PMT)

Returns the future value of a time value of money (TVM) problem. Positive values mean a cash inflow while negative numbers mean a cash outflow. This function is only available if the finance library is installed.

- **PV:** present value. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **PMT:** periodic payment amount. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **I%:** yearly interest rate expressed as a percentage
- **N:** total number of periods

Category: finance

Input Modes: solver

Examples:

tvmfv(600; 2.5; -10000; -200) : returns 273,495.59

tvmfv(N; I%; PV; PMT; P/Y; C/Y; B)

Returns the future value of a time value of money (TVM) problem. Positive values mean a cash inflow while negative numbers mean a cash outflow. This function is only available if the finance library is installed. Same variables as above except

- **P/Y:** payment periods per year. If not included, it is assumed to be 12.
- **C/Y:** interest compounding periods per year. If not included, it is assumed to be 12.
- **B:** payment timing (0 for end of period, 1 for beginning of period). If not included, it is assumed to be 0.

Category: finance

Input Modes: solver

Examples:

tvmfv(600; 2.5; -10000; -200; 12; 12; 0) : returns 273,495.59

5.7 G-H

This section covers functions beginning with the letters G through H.

5.7.1 Get Date in Decimal Format

getdate(date)

Returns the date in dd.mmyyyy format given date type date.

Data Types: date, table

Category: date

Input Modes: solver

Examples:

getdate(today()) : returns today's date (e.g, 1.082003 if today's date of 08/01/03)

getdate(adjDate(01.082003; -15) : returns 17.072003

5.7.2 Get Hours in Decimal Format

hrs(value)

Returns the time in decimal hours given a date type or time in hh.mmssmmm format (3.5 decimal hours is 3 hrs, 30 min).

Data Types: integer, floating point, date

Category: date

Input Modes: solver

Examples:

hrs(today()) : returns the current time (e.g, 17:7989 given approximately 5:47 pm)

hrs(10.4830) : returns 10.8083

5.7.3 Get Hours in HH.MMSS Format

hms(value)

Returns the time in hh.mmssmmm format given a date type or the time in decimal hours (3.5 decimal hours is 3 hrs, 30 min).

Data Types: integer, floating point, date

Category: date

Input Modes: solver

Examples:

hms(10.8083) : returns 10.4830

5.7.4 Get Time in Decimal Format

gettime(date)

This function can only be used within a formula – the returned value cannot be viewed in a template. Returns the time in the format hh.mmssmmm given date type date.

Data Types: date

Category: date

Input Modes: solver

Examples:

gettime(today()) : returns the current time (e.g, 18:0635 given 6:06:35 pm)

5.7.5 Greater Than

valueA > valueB

Returns true if valueA greater than valueB.

Data Types: integer, floating point, date

Category: bool

Input Modes: solver

Examples:

1 > 0.55 : returns true

1 > 5 : returns false

1 > 1 : returns false

5.7.6 Greater Than or Equal To

valueA >= valueB

Returns true if valueA greater than or equal to valueB.

Data Types: integer, floating point, date

Category: bool

Input Modes: solver

Examples:

1 >= 0.55 : returns true

1 >= 5 : returns false

1 >= 1 : returns true

5.7.7 Greatest Common Denominator

gcd(valueA; valueB)

Returns the greatest common integer divisor of valueA and valueB, where $-2^{31} <= valueA, valueB < 2^{31}$.

Data Types: integer, floating point, table.

Category: math

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  gcd(12; -63) : returns 3

  gcd({ {5}; {10} }; { {10}; {4} }) : returns { {5}; {2} }

- Order of Operations and Chain Input Modes

  12 gcd 63 +/- = : returns 3

- RPN Input Mode

  12 ENT 63 ENT +/- gcd : returns 3

5.7.8 History

history

Each time the equals button is selected, a new history item is stored. The software stores the last 10 recorded answers. To recall a value from the history list to the view window, select it.

Category: history

Input Modes: order of operations, chain

5.7.9 Hyperbolic Arc-Cosine

acosh(value)

Returns hyperbolic arc-cosine of value.

Data Types: integer, floating point, table

Category: trig

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode
  acosh(74.2099) : returns 5
- RPN, Order of Operations and Chain Input Modes
  74.2099 acosh : returns 5

5.7.10 Hyperbolic Arc-Sine

asinh(value)

Returns hyperbolic arc-sine of value.

Data Types: integer, floating point, table

Category: trig

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode
  asinh(-1.1752) : returns -1
- Order of Operations and Chain Input Modes
  1.1752 +/- asinh : returns –1
- RPN Input Mode
  1.1752 +/- asinh : returns –1

5.7.11 Hyperbolic Arc-Tangent

atanh(value)

Returns hyperbolic arc-tangent of value.

Data Types: integer, floating point, table

Category: trig

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode
  atanh(-0.7616) : returns -1
- RPN, Order of Operations and Chain Input Modes
  0.7616 +/- atanh : returns –1

5.7.12 Hyperbolic Cosine

cosh(value)

Returns hyperbolic cosine of value.

Data Types: integer, floating point, table

Category: trig

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode
  cosh(5) : returns 74.2099
- RPN, Order of Operations and Chain Input Modes
  5 cosh : returns 74.2099

5.7.13 Hyperbolic Sine

sinh(value)

Returns hyperbolic sine of value.

Data Types: integer, floating point, table

Category: trig

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode
  sinh(1) : returns 1.1752
- RPN, Order of Operations and Chain Input Modes
  1 sinh : returns 1.1752

5.7.14 Hyperbolic Tangent

tanh(value)

Returns hyperbolic tangent of value.

Data Types: integer, floating point, table

Category: trig

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  tanh(1) : returns 0.7616

- RPN, Order of Operations and Chain Input Modes

  1 tanh : returns 0.7616

5.8 I-N

This section covers functions beginning with the letters I through N.

5.8.1 If

if(boolean; expressionA; expressionB)

If boolean is true, evaluate expressionA, otherwise evaluate expressionB.

Category: bool

Input Modes: solver

Examples:

if(A > 0; 100/A; 0) : returns 100 divided by A if A is greater than 0 (true) or 0 if it is equal to or less than 0 (false)

5.8.2 Integer Part

iPart(value)

Returns integer (whole number) part of value.

Data Types: integer, floating point, table

Category: number

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  ipart(5) : returns 5

  ipart(5.25) : returns 5

- Order of Operations and Chain Input Modes

  5 iPart : returns 5

  5.25 iPart : returns 5

- RPN Input Mode

  5 iPart : returns 5

  5.25 iPart : returns 5

5.8.3 Integer, Convert To

toInt(value)

Returns an integer by converting value to an integer between -4e9 and 4e9.

Data Types: boolean, integer, floating point

Category: number

Input Modes: solver

Examples:

toInt(-5.1) : returns -5

5.8.4 Integral

fnInt ("expression"; "variable"; lower; upper)

Uses numerical integration (Gauss-Kronrod) to return the integral of expression with respect to variable, between lower and upper limits.

- **expression:** the expression to analyze. Must be in quotations.
- **variable:** the variable within the expression. Must be in quotations.
- **lower:** lower limit of the range to analyze
- **upper:** upper limit of the range to analyze

Category: calc

Input Modes: algebraic

Examples:

fnInt("x^2"; "x"; –3; 3) : returns 18

5.8.5 Interest Rate

tvmi(N; PV; PMT; FV)

Returns the yearly interest rate of a time value of money (TVM) problem. This function is only available if p1 Finance Lib is installed.

- **PV:** present value. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **FV:** future value. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **PMT:** periodic payment amount. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **N:** total number of periods

Category: finance

Input Modes: solver

Examples:

tvmi(120; -100000; 0; 150000) : returns 4.06

tvmi(N; PV; PMT; FV; P/Y; C/Y; B)

Returns the yearly interest rate of a time value of money (TVM) problem. This function is only available if p1 Finance Lib is installed. Same variables as above except

- **P/Y:** payment periods per year. If not included, it is assumed to be 12.
- **C/Y:** interest compounding periods per year. If not included, it is assumed to be 12.
- **B:** payment timing (0 for end of period, 1 for beginning of period). If not included, it is assumed to be 0.

Category: finance

Input Modes: solver

Examples:

tvmi(120; -100000; 0; 150000; 12; 12; 0) : returns 4.06

5.8.6 Internal Rate of Return

CfoIRR(CFAmntList)

Returns the internal rate of return of the given cash flow. This function is only available if the finance library is installed.

- **CFAmntList:** list containing cash flow amounts where the first element is the initial cash flow.
- **CFFreqList:** list in which each element specifies the frequency of occurrence for a consecutive cash flow amount in CFList.

Category: finance

Input Modes: solver

Examples:

CfoIRR({-5000; 4000; 3000; 3000}) : returns 47.96

CfoIRR(CFAmntList; CFFreqList)

same as above

Category: finance

Input Modes: solver

Examples:

CfoIRR({-5000; 4000; 3000}; {1; 1; 2}) : returns 47.96

5.8.7 Inverse Cumulative Normal Distribution

invNorm(prob)

Returns the value which would generate the given probability for the cumulative standard Normal distribution.

- **prob:** probability

Category: distr

Input Modes: solver

Examples:

invNorm(0.25) : returns -0.6745

invNorm(area; mean; stddev)

same as above except:

- **mean:** mean of the distribution. If not specified, mean is 0. If specified, must be an integer or floating point number.
- **stddev:** standard deviation of the distribution. If not specified, stddev is 1. If specified, must be an integer or floating point number.

Category: distr

Input Modes: solver

Examples:

invNorm(0.25; 0.5; 1) : returns -0.1745

5.8.8 Last

last

In order of operations and chain input modes, returns the last entry recorded in the history list. In RPN input mode, returns the last item pushed on the stack before the last numeric operation.

Category: number

Input Modes: RPN, order of operations, chain

5.8.9 Least Common Multiple

lcm(valueA; valueB)

Returns the least common integer multiple of valueA and valueB, where $-2^{31} <= valueA, valueB < 2^{31}$.

Data Types: integer, floating point, table.

Category: math

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  lcm(12; -63) : returns 252

  lcm([[5]; [10]]; [[10]; [4]]) : returns [[10]; [20]]

- Order of Operations and Chain Input Modes

  12 lcm 63 +/- = : returns 252

- RPN Input Mode

  12 ENT 63 +/- lcm : returns 252

5.8.10 Less Than

valueA < valueB

Returns true if valueA less than valueB.

Data Types: integer, floating point, date

Category: bool

Input Modes: solver

Examples:

1 < 0.55 : returns false

1 < 5 : returns true

1 < 1 : returns false

5.8.11 Less Than or Equal To

valueA <= valueB

Returns true if valueA less than or equal to valueB.

Data Types: integer, floating point, date

Category: bool

Input Modes: solver

Examples:

1 <= 0.55 : returns false

1 <= 5 : returns true

1 < =1 : returns false

5.8.12 Logarithm

log(value)

Returns the base 10 logarithm of value.

Data Types: integer, floating point, table

Category: math

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  log(4) : returns 0.6021

  log(78.36) : returns 1.8941

  log({ {89; 53; 6}; {42; 76; 21} }) : returns { {1.9494; 1.7243; 0.7782}; {1.6232; 1.8808; 1.3222} }

- Order of Operations and Chain Input Modes

  4 log : returns 0.6021

  78.36 log : returns 1.8941

- RPN Input Mode

  4 log : returns 0.6021

  78.36 log : returns 1.8941

5.8.13 Make Date from Decimal Format

makedate(dd.mmyyyy)

This function can only be used within a formula – the returned value cannot be viewed in a template. Converts a date into a date type representing the inputted date.

Data Types: integer, floating point

Category: date

Input Modes: solver

Examples:

makedate(1.082003) : returns 8/1/03 12:00 am

makedate(dd.mmyyyy; hh.mmssmmm)

This function can only be used within a formula – the returned value cannot be viewed in a template. Converts a date and

time into a date type representing the inputted date.

Data Types: integer, floating point

Category: date

Input Modes: solver

Examples:

makedate(20.071969; 22.5600) : returns 7/20/69 3:52 pm

5.8.14 Maximum

max(valueA [; valueB; ...])

If given a series of values or a list, returns the largest value in the list. If given a series of lists of equal length, returns the largest value of each position within the list. [; valueB; ...] is optional.

Data Types: integer, floating point, date, table

Category: stats

Input Modes: solver

Examples:

max(2; 4; 6; 5; 3) : returns 6

max({1; 2; 3; 4; 5}) : returns 5

max({1; 2; 3; 4; 5}; {3; 6; 2; 1; 8}) : returns {3; 6; 3; 4; 8}

5.8.15 Maximum, Function

fMax("expression"; "variable"; lower; upper)

Uses an iterative method to determine the value of an independent variable for which the local maximum of an expression occurs. The possible values for the independent variable are limited to a range bracketed by the provided upper and lower limits. This function uses the "tolerance" constant.

- **expression:** the expression to analyze. Must be in quotations.
- **variable:** the independent variable within the expression. Must be in quotations.
- **lower:** lower limit of the range to analyze
- **upper:** upper limit of the range to analyze

Category: calc

Input Modes: algebraic

Examples:

- Algebraic Input Mode

  fMax("-x^2"; "x"; -3; 3) : returns -0.0001

fMax("expression"; "variable"; lower; upper ; e)

Same as above except:

- **e:** tolerance. This is optional. If it is not included, it defaults to 10^{-4}. Smaller values may lead to greater accuracy, but will cause the calculation to take longer to complete.

Category: calc

Input Modes: algebraic, RPN

Examples:

- Algebraic Input Mode

  fMax("-x^2"; "x"; -3; 3; 0.5) : returns -0.0001

- RPN Input Mode

  "-x^2" ENT "x" ENT -3 ENT 3 fMax : returns -0.0001

5.8.16 Mean

mean(datalist)

Returns the mean of a list or vector.

- **datalist:** a list containing values used in the calculation.

Data Types: table

Category: stats

Input Modes: solver

Examples:

mean({1; 2; 3; 4; 5}) : returns 3

mean(datalist; occlist)

Same as above except:

- **occlist:** a list, the same size as datalist, containing the number of occurrences of each corresponding value in datalist. Note that if occlist is not provided for functions that take occlist as an optional argument, the function treats each entry in datalist as a single occurrence.

Data Types: table, matrix

Category: stats

Input Modes: solver

Examples:

mean({1; 2; 3; 4; 5}; 1) : returns 3

mean({1; 2; 3; 4; 5}; {2; 4; 4; 6; 6}) : returns 3.4545

5.8.17 Median

median(datalist)

Returns the median of a list or vector.

- **datalist:** a list containing values used in the calculation.

Data Types: table

Category: stats

Input Modes: solver

Examples:

median({1; 2; 3; 4; 5}) : returns 3

median(datalist; occlist)

Same as above except:

- **occlist:** a list, the same size as datalist, containing the number of occurrences of each corresponding value in datalist. Note that if occlist is not provided for functions that take occlist as an optional argument, the function treats each entry in datalist as a single occurrence.

Data Types: table, matrix

Category: stats

Input Modes: solver

Examples:

median({1; 2; 3; 4; 5}; 1) : returns 3

median({1; 2; 3; 4; 5}; {2; 4; 4; 6; 6}) : returns 4

5.8.18 Memory

MEM

There are memory locations available for storing and recalling data. Often, these locations are grouped under MEM. Selecting MEM displays a list of functions:

- STO/Store: select to store an item to memory.
- RCL/Recall: select to recall an item from memory.
- Clear: select to clear the memory locations.

See the Using the Calculator : Memory & Storage : Memory Locations section for additional information.

Data Types: boolean, integer, floating point

Category: not applicable

Input Modes: RPN, order of operations, chain

5.8.19 Minimum

min(valueA [; valueB; ...])

If given a series of values or a list, returns the smallest value in the list. If given a series of lists of equal length, returns the smallest value of each position within the list. [; valueB; ...] is optional. RPN input mode can only handle a single list.

Data Types: integer, floating point, date, table

Category: stats

Input Modes: solver

Examples:

min(2; 4; 6; 5; 3) : returns 2

min({1; 2; 3; 4; 5}) : returns 1

min({1; 2; 3; 4; 5}; {3; 6; 2; 1; 8}) : returns {1; 2; 2; 1; 5}

5.8.20 Minimum, Function

fMin("expression"; "variable"; lower; upper)

Uses an iterative method to determine the value of an independent variable for which the local minimum of an expression occurs. The possible values for the independent variable are limited to a range bracketed by the provided upper and

lower limits. This function uses the "tolerance" constant.

- **expression:** the expression to analyze. Must be in quotations.
- **variable:** the independent variable within the expression. Must be in quotations.
- **lower:** lower limit of the range to analyze
- **upper:** upper limit of the range to analyze

Category: calc

Input Modes: algebraic

Examples:

fMin("x^2"; "x"; -3; 3) : returns -0.0001

fMin("expression"; "variable"; lower; upper ; e)

Same as above except:

- e: tolerance. This is optional. If it is not included, it defaults to 10^{-4}. Smaller values may lead to greater accuracy, but will cause the calculation to take longer to complete.

Category: calc

Input Modes: algebraic, RPN

Examples:

- Algebraic Input Mode

  fMin("x^2"; "x"; -3; 3; 0.5) : returns -0.0001

- RPN Input Mode

  "x^2" ENT "x" ENT -3 ENT 3 fMin : returns -0.0001

5.8.21 Modified Internal Rate of Return

CfoMIRR(I%; CFAmntList)

Returns the modified internal rate of return of the given cash flow. This function is only available if the finance library is installed.

- **I%:** periodic interest rate as a percentage.
- **CFAmntList:** list containing cash flow amounts where the first element is the initial cash flow.
- **CFFreqList:** list in which each element specifies the frequency of occurrence for a consecutive cash flow amount in CFList.

Category: finance

Input Modes: solver

Examples:

CfoMIRR(12; {-5000; 4000; 3000; 3000}) : returns 31.53

CfoMIRR(I%; CFAmntList; CFFreqList)

same as above

Category: finance

Input Modes: solver

Examples:

CfoMIRR(12; {-5000; 4000; 3000}; {1; 1; 2}) : returns 31.53

5.8.22 Modulo Division

mod(valueA; valueB)

Returns remainder of valueA divided by valueB.

Data Types: integer, floating point, table.

Category: math

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  mod(5;2) : returns 1

  mod(6.8; 1.67) : returns 0.12

  mod([[68; 42; 33]; [28; 99; 34]]; [[3; 7; 6]; [5; 22; 7]]) : returns [[2; 0; 3]; [3; 11; 6]]

- Order of Operations and Chain Input Modes

  5 mod 2 = : returns 1

  6.8 mod 1.67 = : returns 0.12

- RPN Input Mode

  5 ENT 2 mod : returns 1

  6.8 ENT 1.67 mod : returns 0.12

5.8.23 Multiplication

valueA * valueB

Returns valueA times valueB.

Data Types: integer, floating point

Category: not applicable

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  5 * 63 : returns 315

  8.2 * 32.65 : returns 267.73

- Order of Operations and Chain Input Modes

  5 * 63 = : returns 315

  8.2 * 32.65 = : returns 267.73

- RPN Input Mode

  5 ENT 63 * : returns 315

8.2 ENT 32.65 * : returns 267.73

5.8.24 Natural Logarithm

ln(value)

Returns the natural logarithm of value.

Data Types: integer, floating point

Category: math

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  ln(4) : returns 1.3863

  ln(52.43) : returns 3.9595

- Order of Operations and Chain Input Modes

  4 ln : returns 1.3863

  52.43 ln : returns 3.9595

- RPN Input Mode

  4 ln : returns 1.3863

  52.43 ln : returns 3.9595

5.8.25 Net Future Value

CfoNFV(I%; CFAmntList)

Returns the net future value of the given cash flow. This function is only available if the finance library is installed.

- **I%:** periodic interest rate as a percentage.
- **CFAmntList:** list containing cash flow amounts where the first element is the initial cash flow.
- **CFFreqList:** list in which each element specifies the frequency of occurrence for a consecutive cash flow amount in CFList.

Category: finance

Input Modes: solver

Examples:

CfoNFV(12; {-5000; 4000; 3000; 3000}) : returns 4,352.96

CfoNFV(I%; CFAmntList; CFFreqList)

same as above

Category: finance

Input Modes: solver

Examples:

CfoNFV(12; {-5000; 4000; 3000}; {1; 1; 2}) : returns 4,352.96

5.8.26 Net Present Value

CfoNPV(I%; CFAmntList)

Returns the net present value of the given cash flow. This function is only available if the finance library is installed.

- **I%:** periodic interest rate as a percentage.
- **CFAmntList:** list containing cash flow amounts where the first element is the initial cash flow.
- **CFFreqList:** list in which each element specifies the frequency of occurrence for a consecutive cash flow amount in CFList.

Category: finance

Input Modes: solver

Examples:

CfoNPV(12; {-5000; 4000; 3000; 3000}) : returns 3,098.35

CfoNPV(I%; CFAmntList; CFFreqList)

same as above

Category: finance

Input Modes: solver

Examples:

CfoNPV(12; {-5000; 4000; 3000}; {1; 1; 2}) : returns 3,098.35

5.8.27 Nominal Interest Rate

IntNom(rate; compoundingperiods)

Returns the nominal interest rate. This function is only available if the finance library is installed.

- **rate:** effective annual interest rate. Must be an integer or floating point number.
- **compoundingperiods:** number of interest compounding periods per year. Must be an integer or floating point number greater than or equal to 0. 0 denotes continuous compounding.

Category: finance

Input Modes: solver

Examples:

IntNom(7; 360) : returns 6.767

IntNom(7; 0) : returns 6.766

5.8.28 Not

! valueA

Returns false if valueA is true and returns true if value is false.

Data Types: integer, floating point

Category: bool

Input Modes: solver

Examples:

! (4+1 == 5) : returns false

! (6 < 5) : returns true

5.8.29 Not Equal

valueA <> valueB
valueA != valueB

Returns true if valueA does not equal valueB.

Data Types: boolean, integer, floating point, date

Category: bool

Input Modes: solver

Examples:

5 != 6 : returns true

(5*3) != 15 : returns false

5.9 O-Q

This section covers functions beginning with the letters O through Q.

5.9.1 Occurrences

countx(datalist)

Returns the total number of data points contained in a list.

- **datalist:** a list containing values used in the calculation.

Data Types: table

Category: stats

Input Modes: solver

Examples:

countx({1; 2; 3; 4; 5}) : returns 5

countx(datalist ; occlist)

Same as above except:

- **occlist:** a list, the same size as datalist, containing the number of occurrences of each corresponding value in datalist. Note that if occlist is not provided for functions that take occlist as an optional argument, the function treats each entry in datalist as a single occurrence.

Data Types: table

Category: stats

Input Modes: solver

Examples:

countx({1; 2; 3; 4; 5}; 1) : returns 5

countx({1; 2; 3; 4; 5}; {2; 4; 4; 6; 6}) : returns 22

5.9.2 Or

valueA || valueB

Returns true if valueA is true or valueB is true.

Data Types: boolean

Category: bool

Input Modes: solver

Examples:

(5 == 5) || (4 > 5) : returns true

(5 != 5) || (4 > 5) : returns false

5.9.3 Parentheses

parentheses ( )

Use parentheses to establish precedence when performing calculations. Items within parentheses are evaluated first.

Category: number

Input Modes: solver, order of operations, chain

Examples:

3 + (4 * 5) : returns 23

(3 + 4) * 5 : returns 35

5.9.4 Payback

CfoPbk(CFAmntList)

Returns the period number when the initial investment of a given cash flow will be paid back. This function is only available if the finance library is installed.

- **CFAmntList:** list containing cash flow amounts where the first element is the initial cash flow.
- **CFFreqList:** list in which each element specifies the frequency of occurrence for a consecutive cash flow amount in CFList.

Category: finance

Input Modes: solver

Examples:

CfoPbk({-5000; 4000; 3000; 3000}) : returns 1.33

CfoPbk(CFAmntList; CFFreqList)

same as above

Category: finance

Input Modes: solver

Examples:

CfoPbk({-5000; 4000; 3000}; {1; 1; 2}) : returns 1.33

5.9.5 Percent

value%

In general, valueA% returns the equvalent of valueA / 100. When it is preceded by a value and a math operand, the following occurs in each case:

value% : value / 100
valueA + valueB% : valueA + (valueA * (valueB / 100))
valueA – valueB% : valueA – (valueA * (valueB / 100))
valueA * valueB% : value A * (valueB / 100)
valueA / valueB% : valueA / (valueB / 100)

Data Types: integer, floating point.

Category: math

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  42 % : returns 0.42

  80 + 10 % : returns 88 (80 + 10% of 80)

  80 - 10 % : returns 72 (80 - 10% of 80)

  522 * 63 % : returns 328.86

  24 / 33 % : returns 72.7273

- Order of Operations and Chain Input Modes

  42 % :returns 0.42

  80 + 10 % : returns 88

  80 - 10 % : returns 72

  522 * 63 % = : returns 328.86

  24 / 33 % = : returns 72.7273

- RPN Input Mode

  42 % : returns 0.42

  522 ENT 63 % : returns 328.86

  24 ENT .33 / : returns 72.7273

5.9.6 Periods

tvmn(I%; PV; PMT; FV)

Returns the total number of periods of a time value of money (TVM) problem. This function is only available if the finance library is installed.

- **PV:** present value. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **FV:** future value. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **PMT:** payment amount. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **I%:** interest rate expressed as a percentage

Category: finance

Input Modes: solver

Examples:

tvmn(7.25; 35000; -3000; 0) : returns 12.13

tvmn(I%; PV; PMT; FV; P/Y; C/Y; B)

Returns the total number of periods of a time value of money (TVM) problem. This function is only available if the finance library is installed. Same variables as above except

- **P/Y:** payment periods per year. If not included, it is assumed to be 12.
- **C/Y:** interest compounding periods per year. If not included, it is assumed to be 12.
- **B:** payment timing (0 for end of period, 1 for beginning of period). If not included, it is assumed to be 0.

Category: finance

Input Modes: solver

Examples:

tvmn(7.25; 35000; -3000; 0; 12; 12; 0) : returns 12.13

5.9.7 Permutations

nPr(n; r)

Returns the number of permutations of n taken r at a time. n, r must be integer values where r <= n, 0 <= n, r <= 170. Returned values correspond to n!/(n-r)!

Data Types: integer, floating point, table

Category: prob

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  nPr(8;3) returns 336

  nPr({10; 11; 12}; 2) returns {90; 110; 132}

- Order of Operations and Chain Input Modes

  8 nPr 3 = : returns 336

- RPN Input Mode

  8 ENT 3 nPr : returns 336

5.9.8 Polar to Rectangular Conversion

imag(r; q)

Returns y coordinate given polar coordinates r and q.

Data Types: integer, floating point, table

Category: cmplx

Input Modes: solver

Examples:

- Trig Mode preference set to Radians

imag(5; 4) : returns –3.7840 when Trig Mode set to radians

imag([[3; 4]; [5; 6]]; [1; 2]; [7; 8]]) : returns [[2.5244; 3.6372]; [3.2849; 5.9361]] when Trig Mode set to radians

real(r; q)

Returns the x coordinate given polar coordinates r and q.

Data Types: integer, floating point, table

Category: cmplx

Input Modes: solver

Examples:

real(5; 4) : returns –3.2682 when Trig Mode set to radians

real([[3; 4]; [5; 6]]; [1; 2]; [7; 8]]) : returns [[1.6209; -1.6646]; [3.7695; -0.8730]] when Trig Mode set to radians

5.9.9 Power

valueA ^ valueB

Returns valueA raised to valueB.

Data Types: integer, floating point, table

Category: math

Input Modes: solver, RPN, order of operations, chain

Examples:

- Algebraic Input Mode

  3 ^ 3 : returns 27

  3 ^ -2.2 : returns .0892

  {5; 6; 7} ^ {1; 2; 3} : returns {5; 36; 343}

- Order of Operations and Chain Input Modes

  3 ^ 3 = : returns 27

  3 ^ 2.2 +/- = : returns .0892

- RPN Input Mode

  3 ENT 3 ^ : returns 27

  3 ENT 2.2 ENT +/- ^ : returns .0892

5.9.10 Power of 10

10 ^ value

Returns 10 raised to the power of value.

Data Types: integer, floating point, table

Category: math

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  10 ^ -1.23 returns 0.0589

  10 ^ {4.2; 1.65; 3.96} returns {15848.9319; 44.6684; 9120.1084}

- Order of Operations and Chain Input Modes

  1.23 +/- 10^ returns 0.0589

- RPN Input Mode

  1.23 ENT +/- 10^ : returns 0.0589

5.9.11 Present Value

tvmpv(N; I%; PMT; FV)

Returns the present value of a time value of money (TVM) problem. Positive values mean a cash inflow while negative numbers mean a cash outflow. This function is only available if p1 Finance Lib is installed.

- **FV:** future value. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **PMT:** periodic payment amount. Positive values mean a cash inflow while negative numbers mean a cash outflow.
- **I%:** yearly interest rate expressed as a percentage
- **N:** total number of periods

Category: finance

Input Modes: solver

Examples:

tvmpv(240; 5; 120; 300) : returns -18,293.63

tvmpv(N; I%; PMT; FV; P/Y; C/Y; B)

Returns the present value of a time value of money (TVM) problem. Positive values mean a cash inflow while negative numbers mean a cash outflow. This function is only available if p1 Finance Lib is installed. Same variables as above except

- **P/Y:** payment periods per year. If not included, it is assumed to be 12.
- **C/Y:** interest compounding periods per year. If not included, it is assumed to be 12.
- **B:** payment timing (0 for end of period, 1 for beginning of period). If not included, it is assumed to be 0.

Category: finance

Input Modes: solver

Examples:

tvmpv(240; 5; 120; 300; 12; 12; 0) : returns -18,293.63

5.9.12 Product

prod(list)

Returns the product of elements in list. In other words, each elements is multiplied together. List may be a table or matrix of integer, double or complex data types.

Data Types: table

Category: stats

Input Modes: solver

Examples:

prod({1; 2; 3; 4; 5}) : returns 120

prod({ {1; 2; 3};{3; 4; 5} }) : returns 360

5.9.13 Profitability Index

CfoProf(I%; CFAmntList)

Returns the profitability index of the given cash flow. This function is only available if the finance library is installed.

- **I%:** periodic interest rate as a percentage.
- **CFAmntList:** list containing cash flow amounts where the first element is the initial cash flow.
- **CFFreqList:** list in which each element specifies the frequency of occurrence for a consecutive cash flow amount in CFList.

Category: finance

Input Modes: solver

Examples:

CfoProf(12; {-5000; 4000; 3000; 3000}) : returns 1.62

CfoProf(I%; CFAmntList; CFFreqList)

same as above

Category: finance

Input Modes: solver

Examples:

CfoProf(12; {-5000; 4000; 3000}; {1; 1; 2}) : returns 1.62

5.9.14 1st Quartile

The matematics community has several methods for computing the first and third quartiles. *powerOne Finance* uses Tukey's method, which includes the median in the quartile calculation. This may be different on other calculators.

quartile1(datalist)

Returns the first quartile of a list or vector.

- **datalist:** a list containing values used in the calculation.

Data Types: table, matrix

Category: stats

Input Modes: algebraic

Examples:

quartile1({1; 2; 3; 4; 5}) : returns 2

quartile1(datalist; occlist)

Same as above except:

- **occlist:** a list, the same size as datalist, containing the number of occurrences of each corresponding value in datalist. Note that if occlist is not provided for functions that take occlist as an optional argument, the function treats each entry in datalist as a single occurrence.

Data Types: table, matrix

Category: stats

Input Modes: algebraic, RPN

Examples:

- Algebraic Input Mode

  quartile1({1; 2; 3; 4; 5}; 1) : returns 2

  quartile1({1; 2; 3; 4; 5}; {2; 4; 4; 6; 6}) : returns 2

- RPN Input Mode

  {1; 2; 3; 4; 5} ENT 1 quartile1 : returns 2

  {1; 2; 3; 4; 5} ENT {2; 4; 4; 6; 6} quartile1 : returns 2

5.9.15 3rd Quartile

The matematics community has several methods for computing the first and third quartiles. *powerOne Finance* uses Tukey's method, which includes the median in the quartile calculation. This may be different on other calculators.

quartile3(datalist)

Returns the third quartile of a list or vector.

- **datalist:** a list containing values used in the calculation.

Data Types: table, matrix

Category: stats

Input Modes: algebraic

Examples:

quartile3({1; 2; 3; 4; 5}) : returns 4

quartile3(datalist; occlist)

Same as above except:

- **occlist:** a list, the same size as datalist, containing the number of occurrences of each corresponding value in datalist. Note that if occlist is not provided for functions that take occlist as an optional argument, the function treats each entry in datalist as a single occurrence.

Data Types: table, matrix

Category: stats

Input Modes: algebraic, RPN

Examples:

- Algebraic Input Mode

  quartile3({1; 2; 3; 4; 5}; 1) : returns 4

  quartile3({1; 2; 3; 4; 5}; {2; 4; 4; 6; 6}) : returns 5

- RPN Input Mode

  {1; 2; 3; 4; 5} ENT 1 quartile3 : returns 4

  {1; 2; 3; 4; 5} ENT {2; 4; 4; 6; 6} quartile3 : returns 5

5.9.16 Quotation Marks

" "

Quotation marks are used to denote text strings. Certain functions, such as solving(), fnInt, and nDeriv, require text strings.

Category: calc

Input Modes: solver

5.10 R

This section covers functions beginning with the letter R.

5.10.1 Radians to Degrees Conversion

degrees(value)

Returns degrees equivalent of value radians.

Data Types: integer, floating point, table

Category: trig

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  degrees(2*pi) returns 360

- Order of Operations and Chain Input Modes

  2 * pi = degrees : returns 360

- RPN Input Mode

  2 ENT pi * degrees : returns 360

5.10.2 Random Integer

randInt(lower; upper)

Returns a random integer from a discrete uniform distribution with the specified bounds (lower <= random number <= upper).

- lower: lower limit for the random integer. Must be an integer.
- upper: upper limit for the random integer. Must be an integer larger than lower.

Category: prob

Input Modes: solver

Examples:

randInt(2; 4) : returns single random integer between 2 and 4

randInt(lower; upper; numsimulations)

Same as above except returns a list of random integers:

- numsimulations: number of times to repeat the simulation. Must be an integer greater than 0 if entered.

Category: prob

Input Modes: solver

Examples:

randInt(1; 10; 5) : returns list of 5 random integers, each with a value between 1 and 10

5.10.3 Random Number

rand()

Returns a random floating point number from a uniform distribution within the bounds 0 <= random number <= 1. The number of decimal places is set based on the current decimal setting preference.

Category: prob

Input Modes: solver, RPN, order of operations, chain

Examples:

rand() : returns single random number in the range 0 <= random number <= 1

rand(numsimulations)

Same as above except returns a list of random floating point numbers:

- numsimulations: number of times to repeat the simulation. Must be an integer > 0 if entered.

Category: prob

Input Modes: solver

Examples:

rand(5) : returns list of 5 random numbers in the range 0 <= random number <= 1

5.10.4 Reciprocal

1 / value

Returns 1 divided by value.

Data Types: integer, floating point, table

Category: math

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  1/15 : returns 0.0667

  1/89.56 : returns 0.0112

  1 / { {1; 2}; {3; 4} } : returns { {1; 0.5}; {0.3333; 0.25} }

- Order of Operations and Chain Input Modes

  15 1/x : returns 0.0667

  89.56 1/x : returns 0.0112

- RPN Input Mode

  15 1/x : returns 0.0667

  89.56 1/x : returns 0.0112

5.10.5 Rectangular to Polar Conversion

abs(x; y)

Returns polar coordinate r given rectangular coordinates x and y.

Data Types: integer, floating point, table

Category: number

Input Modes: solver

Examples:

abs(6; 3) : returns 6.7082

abs(12.5; [[1; 2]; [3; 4]]) : returns [[12.5399; 12.6590]; [12.8550; 13.1244]]

angle(x; y)

Returns polar coordinate q given rectangular coordinates x and y.

Data Types: integer, floating point, table

Category: cmplx

Input Modes: solver

Examples:

angle(5; 4) : returns 0.6747 when Trig Mode set to radians

angle([[3; 4]; [5; 6]]; [1; 2]; [7; 8]]) : returns [[0.3217; 0.4636]; [0.9505; 0.92729]] when Trig Mode set to radians

angle(5; 4) : returns 38.6598 when Trig Mode set to degrees

angle([[3; 4]; [5; 6]]; [1; 2]; [7; 8]]) : returns [[18.4349; 26.5651]; [54.4623; 53.1301]] when Trig Mode set to degrees

5.10.6 Root

root(y; x)

Returns x^{th} root of y.

Data Types: integer, floating point, table

Category: math

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  root(27; 3) : returns 3

root(84.5; 5) : returns 2.4287

root({ {41; 53}; {65; 78} }; 3) : returns { {3.4482; 3.7563}; {4.0207; 4.2727} }

- Order of Operations and Chain Input Modes

  27 $^{x}\sqrt{y}$ 3 = : returns 3

  84.5 $^{x}\sqrt{y}$ 5 = : returns 2.4287

- RPN Input Mode

  27 ENT 3 $^{x}\sqrt{y}$: returns 3

  84.5 ENT 5 $^{x}\sqrt{y}$: returns 2.4287

5.10.7 Round

round(value)

Returns value rounded to the displayed number of decimal places.

- **value:** the value to round. Its data type must be integer, floating point, table.

Category: number

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  round() with one parameter always rounds to the nearest integer.

- Order of Operations and Chain Input Modes

  5.1234 rnd with Decimal Setting preference set to 3 : returns 5.123

  52.1235 rnd with Decimal Setting preference set to 3 : returns 52.124

- RPN Input Mode

  5.1234 rnd with Decimal Setting preference set to 3 : returns 5.123

  52.1235 rnd with Decimal Setting preference set to 3 : returns 52.124

round(value; #decimals)

Returns value rounded to the designated number of decimal places.

- **#decimals:** the number of decimal places to round valueA. This is optional, must be an integer number, and must be in the range 0 <= #decimals <= 10. If #decimals is not designated, value will be rounded to the Decimal Setting preference.

Category: number

Input Modes: solver

Examples:

round(5.1234; 3) : returns 5.123

round(5.1235; 3) : returns 5.124

round([[1.234; 2.3]; [3.356789; 4]]; 2) : returns [[1.23; 2.3]; [3.36; 4]]

5.11 S

This section covers functions beginning with the letter S.

5.11.1 Secant

sec(value)

Returns secant of value.

Data Types: integer, floating point, table

Category: trig

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  sec(0.5236) : returns 1.1547 when Trig Mode Preferences set to Radians

  sec(30) : returns 1.1547 when Trig Mode Preferences set to Degrees

- RPN, Order of Operations and Chain Input Modes

  0.5236 sec : returns 1.1547 when Trig Mode Preferences set to Radians

  30 sec : returns 1.1547 when Trig Mode Preferences set to Degrees

5.11.2 Semi-Colon

semi-colon (;)

Used to separate arguments in function calls.

Data Types: n/a

Category: number

Input Modes: solver

Examples:

root(y; x) : separate the y and x variables in the root function call

5.11.3 Sequence Evaluation

seq("expression"; "variable"; begin; end)

Returns list of floating point numbers derived by evaluating the expression with regard to variable.

- **expression:** expression to evaluate. Must be in quotations.
- **variable:** variable within the expression to evaluate. Must be in quotations.
- **begin:** point to start incrementing.
- **end:** point to stop incrementing.

Category: stats

Input Modes: solver

Examples:

seq("a*2"; "a"; 1; 5) : returns {2; 4; 6; 8; 10} (or {1*2; 2*2; 2*3; 2*4; 2*5})

seq("expression"; "variable"; begin; end; step)

Returns list of floating point numbers derived by evaluating the expression with regard to variable. Same as above except:

- **step:** how often to evaluate between begin and end. If end > begin, step must be negative. This is optional. If step is not included, default is 1. If there are not an exact number of increments in the range then the last value calculated will be just before the end value.

Category: stats

Input Modes: solver

Examples:

seq("a^2"; "a"; 1; 11; 3) : returns {1; 16; 49; 100} (or {1^2; 4^2; 7^2; 10^2})

5.11.4 Show

show

Shows all available decimal places.

Data Types: integer, floating point

Category: number

Input Modes: RPN, order of operations, chain

5.11.5 Sigma

sigma("expression"; "variable"; begin; end)

Returns the sum of values derived by evaluating the expression with regard to variable.

- **expression:** expression to evaluate. Must be in quotations.
- **variable:** variable within the expression to evaluate. Must be in quotations.
- **begin:** point to start incrementing.
- **end:** point to stop incrementing.

Category: stats

Input Modes: solver

Examples:

sigma("a*1"; "a"; 1; 5) : returns 15 (or 1 + 2 + 3 + 4 + 5)

sigma("expression"; "variable"; begin; end; step)

Returns the sum of values derived by evaluating the expression with regard to variable. Same as above except:

- **step:** how often to evaluate between begin and end. If end > begin, step must be negative. This is optional. If step is not included, default is 1. If there are not an exact number of increments in the range then the last value calculated will be just before the end value.

Category: stats

Input Modes: solver

Examples:

sigma("a^2"; "a"; 1; 11; 3) returns 166 or (1^2 + 4^2 + 7^2 + 10^2)

5.11.6 Sign

sign(value)

Returns –1 if value is less than 0, 0 if value is 0, or 1 otherwise.

Data Types: integer, floating point, table

Category: number

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  sign(-5.67) : returns –1

  sign(0) : returns 0

  sign({ {0.0001; -9999}; {0; -5} }) : returns { {1; -1}; {0; -1} }

- Order of Operations and Chain Input Modes

  5.67 +/- sign : returns –1

  5 sign : returns 1

- RPN Input Mode

  5.67 ENT +/- sign : returns –1

  0 sign : returns 0

+/–

Either inserts a negative sign (solver input mode) or changes the sign of the value in the view window (order of operations and chain input modes). This software does not differentiate between negate sign and subtract in solver and RPN input modes.

5.11.7 Sine

sin(value)

Returns sine of value.

Data Types: integer, floating point, table

Category: trig

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  sin(30) : returns -0.9880 when Trig Mode Preferences set to Radians

  sin(30) : returns 0.5 when Trig Mode Preferences set to Degrees

- RPN, Order of Operations and Chain Input Modes

  30 sin : returns -0.9880 when Trig Mode Preferences set to Radians

  30 sin : returns 0.5 when Trig Mode Preferences set to Degrees

5.11.8 Single Payment Future Value

spfv(percent; periods)

Returns the future value of a single $1.00 payment. This function is only available if the finance library is installed.

- **percent:** interest rate per compounding period expressed as a percentage.
- **periods:** number of payments.

Category: finance

Input Modes: solver

Examples:

spfv(5; 360) : returns 42,476,396.41

spfv(5/12; 360) : returns 4.4677

spfv(8; 0) : returns 1

spfv(0; 144) : returns 1

5.11.9 Single Payment Present Value

sppv(percent; periods)

Returns the present value of a single $1.00 payment. This function is only available if the finance library is installed.

- **percent:** interest rate per compounding period expressed as a percentage.
- **periods:** number of payments.

Category: finance

Input Modes: solver, RPN

Examples:

sppv(5; 360) : returns 2.354e-8

sppv(5/12; 360) : returns 0.2238

sppv(8; 0) : returns 1

sppv(0; 144) : returns 1

5.11.10 Solving

solving()

Returns the name of the variable currently being solved for. Using solving when creating templates makes it possible to calculate multiple equations in the same template. The variable name should be in quotations. Note: 'Auto-compute' template preferences should be turned off if the template uses this function. See the Creating Templates section for more information.

Category: templates only

Input Modes: solver

Examples:

if (solving()=="valueA"; valueB+valueC-valueA; 5+valueA-valueA) : returns valueB+valueC when calculating valueA or 5 when calculating valueB or valueC. Since valueA is not included in any calculations, setting valueA - valueA offers valueA for calculation but does not use it in the calculation itself.

5.11.11 Square

value ^ 2

Returns value multiplied by 2.

Data Types: integer, floating point, table

Category: math

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  5 ^ 2 : returns 25

  -3 ^ 2 : returns -9 (Preference setting -2^2=-4)

  -3 ^ 2 : returns 9 (Preference setting -2^2=4)

  { {1; 2}; {3; 4} } ^ 2 : returns { {1; 4}; {9; 16} }

- Order of Operations and Chain Input Modes

  5 x^2 : returns 25

  3 +/- x^2 : returns 9

- RPN Input Mode

  5 x^2 : returns 25

  -3 x^2 : returns 9

5.11.12 Square Root

sqrt(value)

Returns the square root of value.

Data Types: integer, floating point, table

Category: math

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  sqrt(9.9) : returns 3.1464

  sqrt({ {9; 36}; {81; 4} }) : returns { {3; 6}; {9; 2} }

- Order of Operations and Chain Input Modes

  9.9 √x : returns 3.1464

- RPN Input Mode

  9.9 √x :returns 3.1464

5.11.13 Stack

There are special functions for manipulating the stack. These functions can be reached by tapping an item pushed onto the stack or choosing category Stack from the function list and then selecting the desired function. If an item is selected on the stack, the function selected will adjust based on that item. If a stack function is selected from the function list the first item on the stack will be the focal point.

- **Drop:** throws out the item.
- **Duplicate** (dup)**:** copies the item into register 0 (view window/entry line), pushing all others up.
- **Move:** removes the item from its location in the stack and places it in register 0 (view window/entry line).
- **Rotate** (rot)**:** moves the stack in a clockwise direction.
- **Rotate Rvrs** (rotr)**:** moves the stack in a reverse or counter-clockwise direction.
- **Swap:** swaps the item with the contents of register 0 (view window/entry line).

Category: stack (or available by selecting an item on the stack)

Input Modes: RPN

5.11.14 Standard Deviation

stdDev(datalist [; occlist])

Returns the sample standard deviation of the list datalist. occlist is an optional argument that can contain a list of occurences corresponding to the values in datalist.

- **datalist:** a list containing values used in the calculation.
- **occlist:** a list, the same size as datalist, containing the number of occurrences of each corresponding value in datalist. occlist is optional for algebraic input mode but required for RPN input mode. Note that if occlist is not provided for functions that take occlist as an optional argument, the function treats each entry in datalist as a single occurrence.

Data Types: table

Category: stats

Input Modes: solver

Examples:

stdDev({1; 2; 3; 4; 5}) : returns 1.5811

stdDev({1; 2; 3; 4; 5}; {2; 4; 4; 6; 6}) : returns 1.3355

stdDevP(datalist [;occlist])

Returns the population standard deviation of the list datalist. occlist is an optional argument that can contain a list of occurences corresponding to the values in datalist.

- **datalist:** a list containing values used in the calculation.
- **occlist:** a list, the same size as datalist, containing the number of occurrences of each corresponding value in datalist. occlist is optional for algebraic input mode but required for RPN input mode. Note that if occlist is not provided for functions that take occlist as an optional argument, the function treats each entry in datalist as a single occurrence.

Data Types: table

Category: stats

Input Modes: solver

Examples:

stdDevP({1; 2; 3; 4; 5}) : returns 1.4142

stdDevP({1; 2; 3; 4; 5}; {2; 4; 4; 6; 6}) : returns 1.3048

5.11.15 Straight Line Depreciation

DepSLBV (C; S; L; M; Y) (Book Value)
DepSLDA (C; S; L; M; Y) (Depreciation Amount)
DepSLDV (C; S; L; M; Y) (Depreciation Value)

Book value returns the book value (depreciable value + salvage value) for the asset at the end of the given year. Depreciation amount returns the amount that the asset depreciated during the given year. Depreciation value returns the remaining total depreciable value for the asset at the end of the given year. All three are calculated using the straight line method of depreciation. This function is only available if the finance library is installed.

- **C:** cost of the depreciable asset
- **S:** salvage value of the depreciable asset
- **L:** life in years of the depreciable asset
- **M:** first month to begin depreciating (1 is January, 12 is December)
- **Y:** year to calculate

Category: finance

Input Modes: solver

Examples:

DepSLBV (150000; 20000; 20; 6; 3) : returns 133,208.33

DepSLDA (150000; 20000; 20; 6; 3) : returns 6,500

DepSLDV (150000; 20000; 20; 6; 3) : returns 113,208.33

5.11.16 Subtraction

valueA – valueB

returns valueA minus valueB.

Data Types: boolean, integer, floating point, date, table.

Category: not applicable

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  9.32 - 4.89 : returns 4.43

  { {15; 22}; {8; 89} } - 6 : returns { {9; 16}; {2; 83} }

- Order of Operations Input Mode

  9.32 - 4.89 = : returns 4.43

- Chain Input Mode

  9.32 - 4.89 = : returns 4.43

- RPN Input Mode

  9.32 ENT 4.89 - : returns 4.43

5.11.17 Sum of the Year's Digits Depreciation

DepSOYDBV (C; S; L; M; Y) (Book Value)
DepSOYDDA (C; S; L; M; Y) (Depreciation Amount)
DepSOYDDV (C; S; L; M; Y) (Depreciation Value)

Book value returns the book value (depreciable value + salvage value) for the asset at the end of the given year. Depreciation amount returns the amount that the asset depreciated during the given year. Depreciation value returns the remaining total depreciable value for the asset at the end of the given year. All three are calculated using the sum of the year's digits method of depreciation. This function is only available if the finance library is installed.

- **C:** cost of the depreciable asset
- **S:** salvage value of the depreciable asset
- **L:** life in years of the depreciable asset
- **M:** first month to begin depreciating (1 is January, 12 is December)
- **Y:** year to calculate

Category: finance

Input Modes: solver

Examples:

DepSOYDBV (150000; 20000; 20; 6; 3) : returns 119,357.14

DepSOYDDA (150000; 20000; 20; 6; 3) : returns 11,400.79

DepSOYDDV (150000; 20000; 20; 6; 3) : returns 99,357.14

5.11.18 Sum of x-Squared

sumX2(datalist)

Returns the sum of the squares of the values in the list datalist.

- **datalist:** a list containing values used in the calculation.

Data Types: table

Category: stats

Input Modes: solver

Examples:

sumX2({1; 2; 3; 4; 5}) : returns 55

sumX2(datalist; occlist)

Same as above except:

- **occlist:** a list, the same size as datalist, containing the number of occurrences of each corresponding value in datalist. Note that if occlist is not provided for functions that take occlist as an optional argument, the function treats each entry in datalist as a single occurrence.

Data Types: table

Category: stats

Input Modes: solver

Examples:

sumX2({1; 2; 3; 4; 5}; 1) : returns 55

sumX2({1; 2; 3; 4; 5}; {2; 4; 4; 6; 6}) : returns 300

5.11.19 Summation

sumX(datalist)

Returns the sum of the values in the list datalist. occlist is an optional argument that can contain a list of occurences corresponding to the values in datalist.

- **datalist:** a list containing values used in the calculation.

Data Types: table

Category: stats

Input Modes: solver

Examples:

sumX({1; 2; 3; 4; 5}) : returns 15

sumX(datalist; occlist)

Same as above except:

- **occlist:** a list, the same size as datalist, containing the number of occurrences of each corresponding value in datalist. Note that if occlist is not provided for functions that take occlist as an optional argument, the function treats each entry in datalist as a single occurrence.

Data Types: table

Category: stats

Input Modes: solver

Examples:

sumX({1; 2; 3; 4; 5}; 1) : returns 15

sumX({1; 2; 3; 4; 5}; {2; 4; 4; 6; 6}) : returns 76

Also see the function Sigma.

5.12 T-Z

This section covers functions beginning with the letter T through Z.

5.12.1 Tangent

tan(value)

Returns tangent of value.

Data Types: integer, floating point, table

Category: trig

Input Modes: solver, RPN, order of operations, chain

Examples:

- Solver Input Mode

  tan(0.7854) : returns 1 when Trig Mode Preferences set to Radians

  tan(45) : returns 1 when Trig Mode Preferences set to Degrees

- RPN, Order of Operations and Chain Input Modes

  0.7854 tan : returns 1 when Trig Mode Preferences set to Radians

  45 tan : returns 1 when Trig Mode Preferences set to Degrees

5.12.2 Today

today()

This function can only be used within a formula – the returned value cannot be viewed in a template. Returns a date type representing current date and time.

Category: date

Input Modes: solver

Examples:

today() : returns today's date and the current time (e.g, 8/1/03 at 6:09 pm)

5.12.3 Total

CfoTot(CFAmntList)

Returns the sum of the given cash flow and is similar to function Summation. This function is only available if the finance library is installed.

- **CFAmntList:** list containing cash flow amounts where the first element is the initial cash flow.
- **CFFreqList:** list in which each element specifies the frequency of occurrence for a consecutive cash flow amount in CFList.

Category: finance

Input Modes: solver

Examples:

CfoTot({-5000; 4000; 3000; 3000}) : returns 5,000

CfoTot(CFAmntList; CFFreqList)

same as above

Category: finance

Input Modes: solver

Examples:

CfoTot({-5000; 4000; 3000}; {1; 1; 2}) : returns 5,000

5.12.4 Uniform Series Future Value

usfv(percent; periods)

Returns the future value of a series of $1.00 payments. This function is only available if the finance library is installed.

- **percent:** interest rate per compounding period expressed as a percentage.
- **periods:** number of payments.

Category: finance

Input Modes: solver

Examples:

usfv(5; 360) : returns 849,527,908.17

usfv(5/12; 360) : returns 832.26

usfv(8; 0) : returns 0

usfv(0; 144) : returns 144

5.12.5 Uniform Series Present Value

uspv(percent; periods)

Returns the present value of a series of $1.00 payments. This function is only available if the finance library is installed.

- **percent:** interest rate per compounding period expressed as a percentage.
- **periods:** number of payments.

Category: finance

Input Modes: solver

Examples:

uspv(5; 360) : returns 20.00

uspv(5/12; 360) : returns 186.28

uspv(8; 0) : returns 0

uspv(0; 144) : returns 144

5.12.6 Variance

var(datalist [; occlist])

Returns the sample variance of the list datalist. occlist is an optional argument that can contain a list of occurences corresponding to the values in datalist.

- **datalist:** a list containing values used in the calculation.
- **occlist:** a list, the same size as datalist, containing the number of occurrences of each corresponding value in datalist. occlist is optional for algebraic input mode but required for RPN input mode. Note that if occlist is not provided for functions that take occlist as an optional argument, the function treats each entry in datalist as a single occurrence.

Data Types: table

Category: stats

Input Modes: solver

Examples:

var({1; 2; 3; 4; 5}) : returns 2.5

var({1; 2; 3; 4; 5}; {2; 4; 4; 6; 6}) : returns 1.7835

varP(datalist [;occlist])

Returns the population variance of the list datalist. occlist is an optional argument that can contain a list of occurences corresponding to the values in datalist.

- **datalist:** a list containing values used in the calculation.
- **occlist:** a list, the same size as datalist, containing the number of occurrences of each corresponding value in datalist. occlist is optional for algebraic input mode but required for RPN input mode. Note that if occlist is not provided

for functions that take occlist as an optional argument, the function treats each entry in datalist as a single occurrence.

Data Types: table

Category: stats

Input Modes: solver

Examples:

varP({1; 2; 3; 4; 5}) : returns 2

varP({1; 2; 3; 4; 5}; {2; 4; 4; 6; 6}) : returns 1.7025

6 Appendix

6.1 Calculator Error Messages

"error" may appear in the view window. The cause may be one of the following:

- Overflow or underflow occurs in the calculation.
- Divide by 0.
- Reciprocal when x = 0.
- Square root when x < 0.
- Factorial when x < -169, x > 169.
- Natural log when x <= 0.
- Permutations when n < 0, r < 0, or r or n is not an integer.
- Combinations when n < 0, r < 0, or r or n is not an integer.
- Used too many levels of parentheses.
- Entered a number outside the range of 1e-308 to 1e308 and
- -1e–308 to –1e308.

6.2 Restricted Data Names

When creating data types, it is recommended that they do not have the following names because they may by overwritten during software use:

arc	l	t
b	m	tolerance
e	n	u
deriv	pi	v
dist	r	w
i	r2	x
integral	R2	y

Also, variables starting with "ds_" (no quotes) may be overwritten.

6.3 Technical Support

For technical support, please visit Infinity Softworks' web site:

www.infinitysw.com/support

The web site contains:

- Extensive database of technical support questions and answers
- Contact telephone numbers, mail and email addresses
- Free add-ons including templates, skins, graphs, data sets and import/export/beam plug-ins
- Training and curriculum materials

6.4 Printing This Manual

While Infinity Softworks does not offer a printed version of this manual, you are welcome to print it yourself.

To print, you will need Adobe Acrobat Reader (.pdf). If this is the Adobe Acrobat Reader version of the manual, select the menu "File" then "Print". The manual is very long. It may be wise to print the sections needed instead of the entire manual or, if your printer supports it, printing multiple pages per sheet.

To download a copt of Adobe Acrobat Reader, visit Adobe's web site:

http://www.adobe.com/reader

6.5 Legal and Disclaimers

Infinity Softworks, Inc., makes no warranty, either expressed or implied, including but not limited to any implied warranties of merchantability and fitness for a particular purpose, regarding any programs or book materials and makes such materials available solely on an "as-is" basis.

In no event shall Infinity Softworks, Inc., be liable to anyone for special, collateral, incidental, or consequential damages in connection with or arising out of the purchase or use of these materials, and the sole and exclusive liability of Infinity Softworks, Inc., regardless of the form of action, shall not exceed the purchase price of this application. Moreover, Infinity Softworks, Inc., shall not be liable for any claim of any kind whatsoever against the use of these materials by any other party.

7 Index

The index is your guide to key words throughout the manual. It starts on the next page

- B -

- C -

- D -

- E -

- F -

- G -

- H -

- I -

- J -

- K -

- L -

- M -

- N -

- O -

- P -

- Q -

- R -

- S -

- T -

- U -

- V -

- W -

- Y -