



Math Suite Millennium
Extras - Build 1, 09.2001
© Copyright 2001, Jason S. Bailey

Non-profit distribution of unmodified, complete package permitted.
Unauthorized distribution of modified packages is prohibited.
Use this software at your own risk!

'Math Suite Millennium' and 'Millennium Math Suite' are registered trademarks of Math Suite Millennium.

Introduction

Helpful Hints

You will find helpful hints, notes and extra information like this throughout this documentation. These helpful hints will always be found on the left margin.

Calculator Extras?

Once the contents of the MSM archive have been extracted, you will find a folder called *extras* inside the Math Suite Millennium directory (or rather where you chose to extract the MSM archive). Inside this folder, you will find a number of extra calculator functions, programs and utilities. Some of these are math programs; some are not.

If you haven't read through the main documentation (manual.pdf), you are highly encouraged to do so before you proceed to install any *extras*.

Requirements

The only main requirement for any of the extras, is that you have MSM installed on your calculator – especially the programs, which absolutely will not run without MSM. It is recommended, however, that you send them to the MSM folder.

Installation

Installation is simple – ensure MSM is already installed, then send the program (files with *.89p* or *.9xp* extensions) to your calculator with a calculator linking cable and software. For example, if you wanted to install *Defaults Professional 4.0* on your TI-89, simply send *defpro.89g* to your TI-89.

If you have questions about sending the proper files to your calculator, check the installation section of the main MSM documentation, or manual. If you still have questions, please email jasonsbailey@yahoo.com.

Program & Function Usage

In the following sections, are instructions and information for each of the extra programs and functions. Each section deals with a single program or function. The actual calculator file name for each program or function is listed in parenthesis next to the program or function name.

Programs & Functions

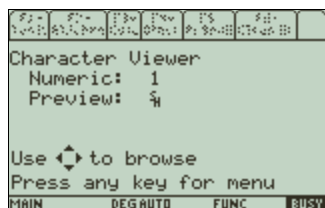
The following are sections which describe how each function or program functions – a complete usage guide.

It is recommended that you read the *Terms & Definitions* section in the main documentation (manual.pdf) before you read further.

Most programs require a single parameter (usually 1). For more information, check the *Program Parameters* section in the main documentation.

Program 1: Character Viewer (charview)

The Character Viewer is a utility that can come in handy for BASIC programmers. Designed to easily browse the calculator's built-in *character set*, and to determine the numerical equivalent to these characters.

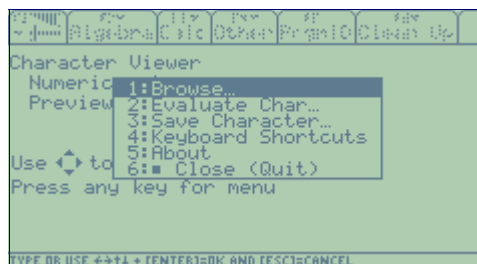


Once the program begins, the *Program I/O* screen is displayed. Near the top of the screen is a numeric value, and the character associated with that value. Essentially, all the program does is increment or decrement a numeric value when the arrow keys are pressed, which it passes to the calculator's

built-in *char()* function, which in-turn, returns a character string (displayed as a *preview*). See the *char()* and *ord()* commands in your calculator manual for more information.

Use the up and down, or left and right arrow keys to increment or decrement the numeric value and change the character *preview*.

To view the main pop-up menu, press any key that isn't a keyboard shortcut (covered below), such as x or y. A number of features are accessible from this menu.



If you wish to close this menu and continue browsing through characters and their numerical equivalents, select *Browse*. If you wish to jump directly to another numerical value, select *Evaluate Char*. Provide a number in the forthcoming dialog box and press Enter. The number and its corresponding character will be displayed.

If you wish to save a character (single character string), select *Save Character*.

Select *Keyboard Shortcuts* to view keyboard shortcuts valid during character browsing. For example, pressing 2nd STO at the browsing screen lets you directly access the *Save Character* feature without using the main pop-up menu. You can also the program at the *Browse* screen

by pressing *Escape* or 2nd, *Quit*.

Select *Close (Quit)* to close the program.

Program 2: Incremental Solver (counter)

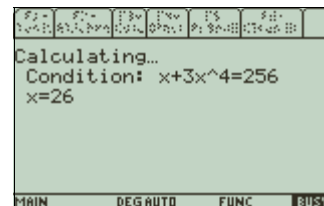
The Incremental Solver is a simple solving program which tests a given condition with values within a given constraint (tested one by one by an incremental loop). Whether this program is actually helpful, is questionable. But it is available if you need it.



Once the program begins, it displays a setup dialog box. In the first text box, type the condition you wish to test (such as $x+3x^4=256$ – it must contain the variable x). In the second box, provide the *step* value. With each loop cycle, the program will advance the

counter value by this value. Finally, provide the start and final values. These will be the start and final counter values (unless the program finds a solution, in which it will terminate before it hits the final value). Press Enter to begin the cycle.

Once the cycle begins, the *Program I/O* screen is displayed. Within it is the condition you are testing, and the current value of x as the counter progresses with each cycle. If a value is found that meets the given condition, the value is displayed in a dialog box. Otherwise, the counter will continue to increment until the final value is reached and a dialog box will appear, announcing an answer wasn't found.



Did you know?

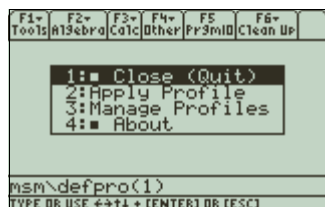
Defaults Professional 4.0 is the direct successor to Defaults Pro 1.0 (there was no version 2 or 3)

Version 4.0 is a complete re-write of the old version, and is much more efficient.

Program 3: Defaults Professional 4.0 (defpro)

Defaults Professional 4.0 is an extremely advanced mode settings management program which allows you to create pre-defined mode setting schemes, or profiles allowing you to set settings globally in one motion.

For example, you could create and customize a profile for your calculus class. Then, you could create and customize another profile for your chemistry class. Using Defaults Professional, you could easily change your settings for each class in a single operation – no need to change each calculator mode setting again and again.



Once the program is started, a pop-up menu will appear. The first selection allows you to close (quit) the program. The second allows you to apply an existing mode setting profile, or scheme. The next selection provides controls for managing profiles. The last provides program information.

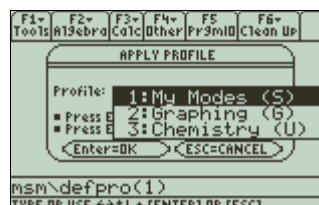
Profile Types

There are several types of profiles supported by the program. Each allow customization of different types of modes or settings. The following is a list of profile types and a description of each:

Type	Description
System	The calculator's main mode settings (Graph, Base, Angle, Split-Screen, Exact/Approx, etc.)
Graph	Graph-specific controls (Graph Order, Axes, Grid, Fields, Solution Method, etc.)
Unit	Custom default units (Length, Volume, Time, Force, Energy, etc.)

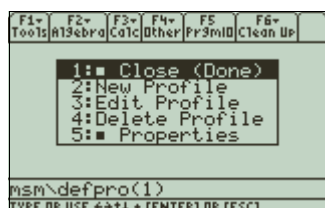
Applying Profiles

Select *Apply Profile* from the main popup menu. A dialog box will appear. A drop-down menu containing a list of current profiles will be present. Next to each profile name is its profile type (S=System, G=Graph, U=Unit).



Select the profile you wish to apply and press Enter. By applying a profile, all applicable settings associated with the profile will be applied (with graph profiles, only modes applicable to the current graph mode will be applied).

Managing Profiles



Select *Manage Profiles* from the main pop-up menu. The profile management pop-up menu will appear. Selections for adding, editing and deleting profiles are visible in this menu. To return to the previous menu, select *Close (Done)*, or press Escape.

New Profiles

To create a new profile, select *New Profile* from the profile management pop-up menu. A dialog box containing a single dialog box will appear. Here, select the type of profile (*System*, *Graph*, or *Unit*) you wish to create and press Enter.

A second dialog box will appear. In the text box, provide a name for your new profile and press Enter. This name will represent your profile when applying, editing or deleting it.

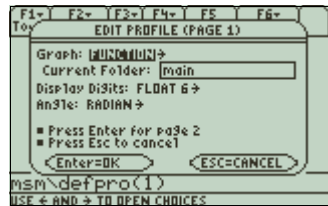


After a short wait, you will be given the chance to customize your profile. The appropriate dialog box, consisting of drop-down menus and text boxes, will be displayed. Press Escape at the first dialog box to return to the profile management pop-up menu and accept the default values for your profile. Or, press Enter to advance to the next dialog box (or page).

Once you have advanced past the first dialog box, pressing the Escape key will return you to the previous dialog box. Your customizations won't be saved until you press Enter at the last dialog box.

Editing Profiles

To edit an existing profile, select *Edit Profile* from the profile management pop-up menu. A dialog box will appear shortly after. Select the profile you wish to edit from the drop-down menu and press Enter.



After a moment, the appropriate dialog box will appear. At this point, press Escape to return to the profile management pop-up screen. Otherwise, begin customizing your profile via the drop-down menus and/or text boxes.

Press Enter to advance to the next screen, or dialog box. You will see additional drop-down menus. Press Escape to return to the previous dialog box. Press Enter at the last dialog box to save your customizations.

Deleting Profiles

Select *Delete Profile* from the profile management pop-up menu. A dialog box will appear, allowing you to select the profile you wish to delete. Press Enter. A second dialog box will ask you to confirm your actions (do you really want to delete the profile?) and press Enter. Once a profile is deleted, there is no way to retrieve it.

Profile Properties

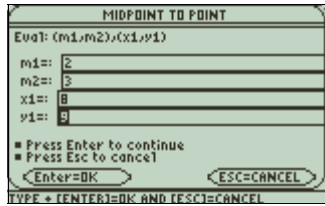
Select *Profile Properties* from the profile management pop-up menu to view/edit a profile's properties. Select the profile you wish to manage in the second dialog box and press Enter. Finally, a dialog box with some information and a few text boxes will be displayed.

Inside the dialog box, you will find the profile's type (System, Graph or Unit), order (placement among other profiles) and name. Although you can't change the profile's type, you can change its order (swap order with another profile) by selecting another value via a drop-down menu. You can also change the profile's name via the text box. Press Enter to save the changes, or press Escape to cancel changes and return to the profile management pop-up menu.



Program 4: Midpoint-to-Point Utility (midpt)

The Midpoint to Point Utility is a simple program which allows you to find the coordinates of the second endpoint of a line segment, when you have the coordinates of one endpoint, and the coordinates of the midpoint (the midpoint is equidistant from both endpoints).

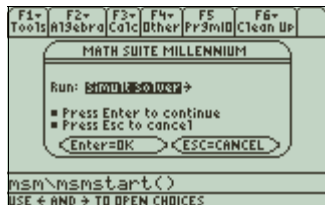


As the program begins, a dialog box is displayed. Given one end point (x_1, y_1) and midpoint (m_1, m_2) , where m_1 is the x value and m_2 is the y value, fill in the text boxes with the appropriate values and press Enter.

In a moment, the second end point is displayed under the program's main toolbar menu. From the toolbar menu, you can save the answers or variables or view an approximation of the answers.

Program 5: MSM Application Launcher (msmstart)

The MSM Application Launcher is for individuals who choose not to install keyboard shortcuts during MSM installation, but would like quick, centralized access to all of MSM's applications.



Simply run the program, and a dialog box appears. Select the program you wish to run from the drop-down menu, and it will run. You can even move the program to the *main* folder and rename it to something like *kbdprgm1*, *kbdprgm2*, etc. so it becomes a keyboard shortcut program.

If you run any MSM program via this launcher, you will get better mode preservation support. If the selected application crashes, the MSM Application Launcher will attempt to restore your old settings (changed when the application began) via the system's exit sequence.

Note that this function does not require a parameter. This is done so you can easily rename it to a keyboard shortcut if you wish (keyboard shortcut programs can't use or require parameters).

Function 1: getprime()

The *getprime()* function returns a list of prime numbers within the given constraints.

Usage:	getprime(start,end)
Example 1:	getprime(1,100)
Example 2:	getprime(123,321)

The *getprime()* function will begin at the value specified by the first parameter, and begin calculating all prime numbers until the end value, specified by the second parameter, is reached.

Function 2: mixednum()

The *mixednum()* function returns a string containing a mixed number of the supplied improper fraction.

Usage:	<code>mixednum(num)</code>
Example 1:	<code>mixednum(34/13)</code>
Example 2:	<code>mixednum(111/33)</code>

The first and only parameter to the *mixednum()* function is a numeric value, and should be an improper fraction. The result is a string containing a mixed number (the whole and fractional portions of the number separated by a space). For example, if the value 34/13 were entered, it would return "2 8/13" – which would mean that 2 is whole, and 8/13 is the fractional portion.

Function 3: **polyang()**

The *polyang()* function returns either the number of polygonal angles based on the sum of the polygonal angles, or the sum of the polygonal angle measures based on the number of polygonal angles.

Usage:	<code>polyang(num,format)</code>
Example 1:	<code>polyang(540,0)</code>
Example 2:	<code>polyang(5,1)</code>

The first parameter should be a numeric value representing either a sum of polygonal angles, or a number of polygonal angles, depending on the second parameter.

The second parameter is the format in which you wish to display the results. There are only two possible results:

<i>Value</i>	<i>Description</i>
0	The function will take the first parameter as a polygonal angle sum and attempt to calculate the number of sides the polygon contains
1	The function will take the first parameter as a number of polygonal angles and will attempt to calculate the polygonal angle sum.

For example, if you wanted to know how many angles a polygon with a polygonal angle sum of 540 had, you'd enter *polyang(540,0)*. On the other hand, if you wanted to know the polygonal angle sum of a polygon with 5 angles, you'd enter *polyang(5,1)*.

Function 4: **rdiv()**

The *rdiv()* function performs remainder division and returns a list or expression containing the dividend and remainder, depending on the second parameter.

Usage:	<code>rdiv(frac,format)</code>
Example 1:	<code>rdiv(11/3,1)</code>

Usage:	<code>rdiv(frac,format)</code>
Example 2:	<code>rdiv(27/8,0)</code>

The first parameter of the *rdiv()* function is the fraction you wish to divide. The second parameter is the format in which to return the results. The following are valid values for the format parameter (0,1):

<i>Value</i>	<i>Description</i>
0	The function will return the dividend and remainder as a string seperated by the letter "r".
1	The function will return the dividend and remainder as a list, where the first list item is the dividend, and the last list item is the remainder.

For example, *rdiv(11/3,0)* would return "3r2", and *rdiv(11/3,1)* would return {3,2}.