

MathScript

Simon Ihmig

COLLABORATORS

	<i>TITLE :</i> MathScript		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Simon Ihmig	October 21, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	MathScript	1
1.1	MathScript Help	1
1.2	Introduction	1
1.3	System requirements	2
1.4	Starting the program	2
1.5	First Steps	2
1.6	The main window	2
1.7	Writing the formula	2
1.8	Preview of the formula	3
1.9	Formatting the formula	3
1.10	objects	3
1.11	The Selection Gadgets	3
1.12	Load, Save, Export	4
1.13	Main Window	4
1.14	toolbar	4
1.15	zoom	5
1.16	Selection Gadgets	5
1.17	Operators	5
1.18	Arrows	5
1.19	Logic	5
1.20	Brackets	5
1.21	sum,product und integral	5
1.22	small greek characters	5
1.23	big greek characters	6
1.24	Equations	6
1.25	symbols of the theory of sets	6
1.26	Symbols	6
1.27	Super-/subscript	6
1.28	Fraction and root	6
1.29	derivatives	6

1.30	Matrices	6
1.31	Matrix Window	7
1.32	Input gadget	7
1.33	Preview area	7
1.34	Quick Help	7
1.35	Settings Window	7
1.36	System Settings	7
1.37	General Settings	8
1.38	ARexx Settings	8
1.39	Export Settings	8
1.40	EPS Settings	8
1.41	Binary EPS Settings	8
1.42	IFF Settings	9
1.43	TIFF Settings	9
1.44	Reset to Defaults	9
1.45	Last Saved	9
1.46	Restore	9
1.47	menus	9
1.48	new	9
1.49	load	9
1.50	save	10
1.51	Save as	10
1.52	import	10
1.53	export	10
1.54	Graphic formats	10
1.55	eps	10
1.56	Binary EPS	11
1.57	iff	11
1.58	tiff	11
1.59	About	11
1.60	quit	11
1.61	Execute ARexx script	11
1.62	Function keys	11
1.63	Change settings	11
1.64	Load settings	11
1.65	Save settings	12
1.66	Save as settings	12
1.67	arexx	12
1.68	clear	12

1.69	getcontents	12
1.70	setcontents	13
1.71	getposition	13
1.72	setposition	13
1.73	insert	13
1.74	Load	13
1.75	Save	13
1.76	Import	13
1.77	Export	13
1.78	screenToFront	13
1.79	requester	14
1.80	getSystemPrefs	14
1.81	setSystemPrefs	14
1.82	getGeneralPrefs	14
1.83	setGeneralPrefs	14
1.84	getArexxPrefs	14
1.85	setArexxPrefs	14
1.86	getExportPrefs	15
1.87	setExportPrefs	15
1.88	SavePrefs	15
1.89	SavePrefsAs	15
1.90	LoadPrefs	15
1.91	previewPrefs	15
1.92	GetDimensions	15
1.93	SetZoom	15
1.94	GetZoom	16
1.95	codes	16
1.96	Disclaimer	17
1.97	Registration	17
1.98	Registration form	17
1.99	Author	18
1.100	Future	18
1.101	credits	18
1.102	mui	18

Chapter 1

MathScript

1.1 MathScript Help

MathScript © Copyright Simon Ihmig 1994,1995 MathScript is SHAREWARE !

Guide for MathScript

1. Introduction

1.1 **Description** - What is it? 1.2 **System requirements** - Will it run on my system?

2. Using the program

2.1 **Starting the program** - How to start it from CLI or WB 2.2 **First Steps** - For Beginners 2.3 The User Interface 2.3.1 **Main Window** - Explanation of the main GUI elements 2.3.2 **Settings Window** - How to change the default settings 2.3.3 **Menus** - List of all menu items

3. Appendix

3.1 **ARexx port** - For experts 3.2 **Control codes** - Essential list of all control codes

Disclaimer - For all those lawyers of you **Registration** - MathScript is shareware. Please register! **Author** - That's me **The future** - Upcoming features **Credits** - Thanks go to... **MUI** - The Magic User Interface

1.2 Introduction

MathScript is a formula editor, which is used to integrate mathematical formulas into word processors. In principle this is nothing new but until now it was a deficit for the Amiga. Although it is possible to create such formulas with TeX, it is not very convenient.

That is why MathScript was created.

With this program it is very easy to create complex formulas. Use of vector fonts guarantees high quality. To insert the formulas into word processors or DTP programs they are saved as graphics files.

These are the features of version 2.1:

- extensive amount of mathematical and physical symbols
- Control codes for formatting (super-, subscript, fractions, roots)
- easy input with **MUI**
- Export in the following formats: EPS Binary EPS (EPS with TIFF preview IFF_ILBM TIFF
- Import of exported files
- Preview of the formula in any resolution
- adjustable fore- and background color for exported files.
- Online Help (AmigaGuide + GadgetHelp)
- ARexx port
- ARexx scripts on function keys
- support for the locale.library (beginning with OS 2.1). Available languages: english,deutsch,français,svenska,suomi,dansk. You may also create your own languages (see Catalogs/readme).
- AppWindow/AppIcon support
- Automatic generation of icons

1.3 System requirements

Minimum:

OS 2.0 **MUI** 2.2 1MB RAM

Recommended:

OS 2.1, better: 3.x **MUI** 2.3 1.5MB RAM 68020 CPU

Furthermore the post.library is needed. It is included in this software package.

1.4 Starting the program

Starting MathScript ...

from the Workbench:

To start MathScript from the Workbench enter the directory where the program is installed and double click its icon. Alternatively you can click on an icon of a previously saved formula to start MathScript loading this formula automatically.

from the Shell:

To launch MathScript from the Shell type in "MathScript:MathScript". The following arguments are accepted: FILE

FILE : name of a formula which shall be loaded immediately

1.5 First Steps

This chapter offers you a short introduction to MathScript. If you are not familiar with it, read this chapter page by page. While reading about a topic it is the best to practice it with the program at once. Therefore start MathScript.

Let's go!

1.6 The main window

The main window

After you have started MathScript the **main window** will appear. It is subdivided into four parts:

1. The **Toolbar** 2. The **Selection gadgets** 3. The **Input gadget** 4. The **Preview area**

We will first concentrate on the input gadget.

[Next page](#)

1.7 Writing the formula

Writing the formula

For writing the formulas into MathScript the **input gadget** is used. The input is made with the keyboard or with the **selection gadgets**. Formatting functions are specified by special **control codes**, thus without WYSIWYG representation.

To enter standard ASCII letters the keyboard is used while the input gadget is active.

We will now create our first formula. Type the following string into the input gadget: "x=1/2" (without quotation marks!)

[Next page](#)

1.8 Preview of the formula

Preview of the formula

Now we want to show the formula. We use the **Preview area** for that purpose. Click into the big empty area. By that the preview is activated. Until the formula is shown, you may have to wait some seconds. The first time you show the preview, it will take more time because the PostScript font has to be loaded first.

In our example the following should appear in the preview area: 1×-2

[Next page](#)

1.9 Formatting the formula

Formatting the formula

In order to format a formula, which means to change the size and position of certain parts, **Control codes** are used. In our first example we already have used such a control code: the character `'/'`. It causes a fraction.

Its Syntax is:

Fraction: syntax: `<Obj1>/<Obj2>` function: Creates a fraction. input: `<Obj1>` - nominator `<Obj2>` - denominator

Each control codes needs **objects**. Following the syntax above, this control code needs two objects, namely a nominator and a denominator. Thus in our example `<Obj1>` was `'1'` and `<Obj2>` was `'2'`.

A complete overview of all supported control codes together with their syntax can be found in **control codes**.

[Next page](#)

1.10 objects

The **control codes** need objects. Such an object is in the simplest case a single character.

So writing `"1/2x"` results in the following illustration: $1 - x^2$ But if you wish the denominator to be $2x$, thus using more than one character, you have to group it to a single object. This is done by surrounding the concerning characters by brackets. So `1/[2x]` causes the following term: $1 - 2x$ Of course it is possible to create a grouped object inside of another grouped object. To create a double fraction simply write `[1/x]/3` which results in the following: $1 - x - - 3$

[Next page](#)

1.11 The Selection Gadgets

The Selection Gadgets

Although one can use all **control codes** with the keyboard, it is not very convenient to have all of them in mind. Moreover MathScript has a variety of special mathematical and physical symbols, which are not available by the keyboard. To use such symbols and control codes in an easy way, the **Selection Gadgets** are used.

There are 13 such gadgets. If you click on one of them, a window will pop up below, which is divided into single fields. These fields are covered with different symbols. If you keep the mouse button pressed you can select one of them. When you release the button the corresponding symbol will be inserted into the **input gadget**. If you do not want to select one, simply release the button if no one is selected. Furthermore you can control these gadgets with the keyboard. Use the tab key to select the active gadget, press return and use the cursor keys to highlight the desired symbol, then press return again to insert it, press space to close the menu without insertion. You can also use the keys `<right Amiga>` `<1>` - `<6>` to pop up the lower gadgets and `<right Amiga>` `<shift>` `<1>` - `<7>` to pop up the gadgets above.

Beside simple characters **control codes** can be inserted, too. They are shown in a symbolic representation and not in the way they are written. The **objects** belonging to them are symbolized in the following manner: \square stands for a grouped object, # for a single character.

Example: The symbol # means superscript. It inserts the following text into the **input gadget**: x^{\square} . Lets's suppose there is a single x standing behind the cursor. After clicking on the symbol the contents of the input gadget becomes like this: x^{\square} . The cursor is placed automatically behind the first grouped **object**, so behind the ' \square '. By pressing the key '2' the text is " $x^{\square 2}$ ", which leads to this "picture": x^2 . But if the first object is a grouped one, you have to select its gadget first and type in its text afterwards in order to be placed between the brackets (see **objects**).

[Next page](#)

1.12 Load, Save, Export

Load, Save, Export

You should be able to create rather complex formulas now. If you have created one and it looks in the preview like you have imagined, you should save it now. Therefore select the menu item **Save as...** in the "Project" menu. If you have already saved or loaded a formula, you can save the current under the same name by selecting the menu item **Save**. With the menu item **Load...** you can load a previously saved formula. The same is possible by dragging an icon of a formula over the **main window**.

So called FML files are used by Load/Save. These are only useable with MathScript and contain only the formula description which is written in the **input gadget**. In order to use the formulas in other applications, they have to be exported in a **graphic format**. Therefore use the menu item **Export....** It contains additional submenu items with which you can chose the **graphic format**.

Well, that's it! You should have acquired the basic skills to use MathScript. The following chapters go more into details.

Back to the [Contents](#).

1.13 Main Window

The Main Window is subdivided into the following regions:

[Toolbar](#)

[Selection Gadgets](#)

[Input gadget](#)

[Preview area](#)

[Quick Help](#)

1.14 toolbar

You can access the most wanted menu items with this toolbar, namely:

[New Load Save](#)

[Import Export](#)

[Start ARexx script](#)

[Change Settings](#)

[Zoom](#)

1.15 zoom

Here you can set the zoom level of the **Preview**. With this it is possible to change the size of the preview without changing the settings. All numbers are in percent (max. 3 digits). With the popup gadget you can select some of the most used zoom levels.

1.16 Selection Gadgets

There are the following selection gadgets:

operators arrows logic brackets sum,prod,int greek chars GREEK CHARS
equations theory of sets symbols super-/subscript fraction,root derivatives Matrices

1.17 Operators

Contains a collection of mathematical operators.

Keyboard shortcut: '`<right Amiga> <shift> 1`'

1.18 Arrows

There are 5 single and 5 double arrows.

Keyboard shortcut: '`<right Amiga> <shift> 2`'

1.19 Logic

Holds logical symbols.

Keyboard shortcut: '`<right Amiga> <shift> 3`'

1.20 Brackets

Includes **control codes** for making brackets of variable size. There are for each bracket type (`{`,`(`,`[`,`]`,`|`,`<`) left and right sided versions and brackets on both sides.

Keyboard shortcut: '`<right Amiga> <shift> 4`'

1.21 sum,product und integral

Three characters symbolizing a sum, product and an integral.

Keyboard shortcut: '`<right Amiga> <shift> 5`'

1.22 small greek characters

Contains all small greek characters.

Keyboard shortcut: '`<right Amiga> <shift> 6`'

1.23 big greek characters

Includes all greek characters.

Keyboard shortcut: '`<right Amiga> <shift> 7`'

1.24 Equations

Holds the most important equation and inequality symbols.

Keyboard shortcut: '`<right Amiga> 1`'

1.25 symbols of the theory of sets

A collection of symbols of the theory of sets.

Keyboard shortcut: '`<right Amiga> 2`'

1.26 Symbols

Includes symbols not belonging to other groups.

Keyboard shortcut: '`<right Amiga> 3`'

1.27 Super-/subscript

Contains all **control codes** for positioning **objects**.

Keyboard shortcut: '`<right Amiga> 4`'

1.28 Fraction and root

Holds **control codes** for making fractions and roots.

Keyboard shortcut: '`<right Amiga> 5`'

1.29 derivatives

Contains control codes for creating derivatives and vectors.

Keyboard shortcut: '`<right Amiga> 6`'

1.30 Matrices

Several **control codes** for different matrices. The last symbol opens the **matrix window**, which is used to insert any (m,n)-matrix.

1.31 Matrix Window

You can chose any matrix in this window.

Rows Enter the number of rows. Columns Enter the number of columns.

Click on "Ok" to insert this kind of matrix, else "Cancel".

1.32 Input gadget

All formulas are typed into the input gadget. It is a standard string gadget and therefore accepts all key short cuts, which can be applied to other string gadgets.

Press <space> to activate it.

1.33 Preview area

In this area the preview of the current formula is displayed. In order to generate such a preview just click on this area.

It may happen that the full formula cannot be displayed. Use the scroller gadgets to adjust the visible area.

1.34 Quick Help

When you point your mouse over a gadget of the **Main Window** a short help text will be displayed here.

1.35 Settings Window

The settings window can be opened by selecting the menu item **Change settings**.

It is divided into four different "pages", namely:

System

General

ARexx

Export

Moreover there is a seperate menu with the following items: **Reset to Defaults Last Saved Restore**

In order to leave the settings window, select "Ok" or "Cancel".

1.36 System Settings

On this page you can set some options regarding the (operating) system.

Path With the small filerequester gadget or the string gadget you can specify a path MathScript uses as a default for loading/saving.

No Icons If this option is not active MathScript will create suitable icons while saving/exporting. By replacing the icons in the drawer "icons" you can use your own icons.

1.37 General Settings

On this page you can set general preferences.

Font size You are able to specify the font size in pt (1/72 inch) here.

Preview resolution Both gadgets can be used to set the x and y resolution of the preview in dpi (dots per inch).

Colors You are able to specify the fore- and background colors here. There are two pages for each color. OS 2.x users will get three slider gadgets there, while OS 3.x users will get a colorwheel. The colors are not used for the **Preview**, but for the exported files. Note: The EPS based formats do not support the background color.

1.38 ARexx Settings

You can place some ARexx scripts on your function keys on this page.

Function keys For each function key Fn (n=1,2,...,10) you can specify an ARexx script in the string gadgets, which will be executed by pressing this key or selecting the corresponding menu item **Fn:** Pressing the PopUp gadgets on the right side will open an filerequester.

Only names You can specify wheter an ARexx script should appear in the "Arexx" menu only with its name or with the full path.

1.39 Export Settings

You can set some options for the different **graphic formats** on this page. It consists of some other pages which contain settings for a specific format, namely:

EPS

Binary EPS

IFF

TIFF

1.40 EPS Settings

There is only one option:

Include font If this option is activated the postscript description of the math font is embedded into the EPS files. This is necessary if the postscript interpreter is unable to load fonts from your harddisk. This applies of course to all postscript printers. But if a application with a software interpreter (Post, PostView, FinalWriter...) is used, it will look (with a suitable init.ps file) in the directory PSFonts: after available fonts. So in this case it is better not to include the font, so the size of your EPS files will not grow too much. If you want to use your formula on another computer you should activate this option, because the font will not be available on other systems.

1.41 Binary EPS Settings

There are the following options regarding the "Binary EPS" format:

Include font If this option is activated the postscript description of the math font is embedded into the EPS files. This is necessary if the postscript interpreter is unable to load fonts from your harddisk. This applies of course to all postscript printers. But if a application with a software interpreter (Post, PostView, FinalWriter...) is used, it will look (with a suitable init.ps file) in the directory PSFonts: after available fonts. So in this case it is better not to include the font, so the size of your EPS files will not grow too much. If you want to use your formula on another computer you should activate this option, because the font will not be available on other systems.

Preview resolution Both gadgets are used to specify the resolution in dpi of the preview part of the EPS files.

1.42 IFF Settings

There is only one option available regarding the IFF format:

Resolution Both gadgets are used to specify the x and y resolution in dpi.

1.43 TIFF Settings

There is only one option available regarding the TIFF format:

Resolution Both gadgets are used to specify the x and y resolution in dpi.

1.44 Reset to Defaults

The default settings will be used.

1.45 Last Saved

The last saved settings will be loaded again.

1.46 Restore

The settings will be set to the state they had after opening the **Settings Window**.

1.47 menus

Overview of all menu items in the **Main Window**:

Project **New Load ... Save Save as ... Import... Export Graphic formats About ... Quit**

ARexx **Execute ARexx script ... F1: ... F2: F10: ...**

Settings **Change settings... Load settings... Save settings Save as settings...**

1.48 new

Clears the **input gadget** and the **Preview**.

Warning: This action cannot be made undone.

1.49 load

Opens a file requester in which a file can be chosen which should be loaded. Only files which have the suffix .fml are shown there. It is appended automatically by the **Save** function. If you wish to load an ASCII file without this suffix, just clear the gadget "Pattern".

You can also use MathScript's AppWindow/AppIcon support. Simply drag an icon of a formula over the **main window** or AppIcon (if iconified).

1.50 save

Does the same as **Save as ...**, except that if a file has been loaded or saved before no file requester will pop up and the current file will be used for saving.

The name of the current file is shown on the title bar.

1.51 Save as

This saves the current contents of the **input gadget** as a Formula file, in order to reuse or modify the formula later on. A file requester will be opened, in which a file can be chosen, which should be used for saving.

All formulas saved with MathScript have the suffix .fml, in order to distinguish them from regular text files.

1.52 import

When a formula is **exported** its formula description is included into the graphics file. With this function it is possible to load it from a previously exported formula.

If an exported file was modified in any way by another program, the formula description may not be found. In this case you are able to search the whole file for it. For example a word processor may include an EPS file into an IFF file when copying it into the clipboard. With this feature you can search the clipboard file for the formula.

This feature is supported by all **graphics formats** except **EPS**.

1.53 export

This function saves the current formula as a graphics file which can be loaded into other applications (word processors,DTP programs,...).

MathScript supports different **graphics formats** which are selectable as submenu items.

1.54 Graphic formats

In the current version MathScript supports the following formats:

EPS Binary EPS IFF TIFF

1.55 eps

This format is the most popular vector graphic format. In opposition to bitmap formats like the IFF-ILBM they have an excellent printing quality.

EPS means Encapsulated PostScript. PostScript is a interpretative page describing language, which is known by several laser printers. If you do not have such a printer, you are dependent on the software, in which the formulas should be used. For example the word processor "FinalWriter" is able to convert EPS files to Bitmaps so they can be printed on regular printers..

This format contains only Ascii characters so it can be send directly to a PostScript printer or displayed by a software interpreter like "Post". If this is not neccessary you should use the variation **Binary EPS**.

This format does not include the formula description, so the menu item **Import** does not work with these files.

1.56 Binary EPS

This format contains the regular **EPS** code and a bitmap in the **TIFF** format as a preview.

The advantage of this format against the pure Ascii **EPS** is that it can be easily shown on the monitor and printed out on non-PostScript printers.

1.57 iff

There is not much to say about the IFF-ILBM format, is it? ;)

Because it is a bitmap format you will not reach the printing quality of (Binary) **EPS** files. On the other side it should be loadable by all graphic supporting programs.

1.58 tiff

The TIFF format is bitmap based. It has not the importance on the Amiga like the **IFF** format, but it was needed for the **Binary EPS** format. Nevertheless it could be useful when using MathScript formulas on other computer systems because it is very widespread.

1.59 About

Opens a window, in which there are informations about the program and me.

1.60 quit

Quits the program.

1.61 Execute ARexx script

Opens a file requester in which an ARexx script can be chosen which should be executed. Detailed informations about the ARexx port of MathScript can be found under **ARexx**.

1.62 Function keys

In the **Settings Window** on the ARexx page there are the Function keys gadgets. If you have installed an ARexx scripts on a particular function key, you are able to execute that script by selecting the corresponding menu item, too.

1.63 Change settings

This opens the **Settings Window** in which MathScript's settings can be changed.

1.64 Load settings

You can load the settings out of a previously saved prefs file with this menu item.

1.65 Save settings

This will save your settings to be used next time.

1.66 Save as settings

Selecting this menu item you are able to save the current settings into any file.

1.67 arexx

With its ARexx port MathScript can be controlled from "outside" what means without intervention of the user.

MathScript allocates for each running task its own ARexx port. The name of the first one is "MATHSCRIPT.1", the second is called "MATHSCRIPT.2" and so on.

An ARexx script can be started by MathScript with the menu item **Execute ARexx script**.

Arguments follow the standard Amiga ReadArgs() conventions.

Here is the list of all supported commands:

Clear Export

GetARexxPrefs GetContents

GetDimensions GetExportPrefs

GetGeneralPrefs GetPosition

GetSystemPrefs GetZoom

Import Insert

Load LoadPrefs

Preview Requester

Save SavePrefs

SavePrefsAs ScreenToFront

SetARexxPrefs SetContents

SetExportPrefs SetGeneralPrefs

SetPosition SetSystemPrefs

SetZoom

1.68 clear

Clear: syntax: Clear function: Clears the **input gadget** corresponding the menu item **New**. input: - result: -

1.69 getcontents

GetContents: syntax: GetContents function: Provides the current contents of the **input gadget**. input: - result: result - contents

1.70 setcontents

SetContents: syntax: SetContents CONTENTS/A/F function: Sets the new contents of the **input gadget**. input: CONTENTS - New contents of the input gadget result:

1.71 getposition

GetPosition: syntax: GetPosition function: Gets the position of the cursor in the **input gadget** input: result: result - position of the cursor

1.72 setposition

SetPosition: syntax: SetPosition POSITION/A/N function: Sets the cursor on a new position. input: POSITION/A/N - new position of the cursor result:

1.73 insert

Insert: syntax: Insert STRING/A/F function: Inserts a string behind the current cursor position into the input gadget. input: STRING - string that should be inserted result: -

1.74 Load

Load: syntax: Load FILE,FORCE/S function: Loads a text file as a formula. If FILE is not supplied a file requester will pop up. input: FILE - name of the file which should be loaded FORCE - surpresses any warnings result: -

1.75 Save

Save: syntax: Save FILE,FORCE/S function: Saves the current formula as a text file. If FILE is not supplied a file requester will pop up. input: FILE - name of the file which should be saved FORCE - surpresses any warnings result: -

1.76 Import

Import: syntax: Import FILE,FORCE/S function: Searches a file for a formula description. If FILE is not supplied a file requester will pop up. input: FILE - name of the file which should be loaded FORCE - surpresses any warnings result: -

1.77 Export

Export: syntax: Export FORMAT/A,FILE,FORCE/S function: Saves the current formula as a graphic file. If FILE is not supplied a file requester will pop up. input: FORMAT - Format of the file. Possible values: "EPS","BINEPS","IFF","TIFF". FILE - name of the file which should be saved FORCE - surpresses any warnings result: -

1.78 screentofront

ScreenToFront: syntax: ScreenToFront function: Moves the screen where MathScript has opened its windows to the front. input: - result: -

1.79 requester

Requester: syntax: Requester TITLE/A,PATTERN/K function: Opens a file requester for loading files. input: TITLE - Titlebar of the requester PATTERN - Amiga-Dos pattern, which can be used for hiding files result: -

1.80 getsystemprefs

GetSystemPrefs: syntax: GetSystemPrefs ITEM/A function: Gets the current system settings input: ITEM - one of the following values: PATH - default path NOICONS - returns YES or NO result: value corresponding to ITEM

1.81 setsystemprefs

SetSystemPrefs: syntax: SetSystemPrefs PATH/K,NOICONS/K,PROMPT/S function: Sets the current system settings input: PATH - default path NOICONS - YES or NO PROMPT - If this argument is specified, all other arguments are ignored and the **Settings Window** will be opened result: value corresponding to ITEM

1.82 getgeneralprefs

GetGeneralPrefs: syntax: GetGeneralPrefs ITEM/A function: Gets the current general settings input: ITEM - one of the following values: FONTSIZE - Font size in pt PREVIEW_DPI - returns x and y dpi of the preview seperated by a 'x' FG - Foreground color, format: "r,g,b" BG - Background color, format: "r,g,b" result: value corresponding to ITEM

1.83 setgeneralprefs

SetGeneralPrefs: syntax: SetGeneralPrefs FONTSIZE/K,N,PREVIEW_DPI/K,FG/K,BG/K,PROMPT/S function: Gets the current general settings input: FONTSIZE - Font size in pt PREVIEW_DPI - x and y dpi of the preview seperated by a 'x' FG - Foreground color, format: "r,g,b" BG - Background color, format: "r,g,b" PROMPT - If this argument is specified, all other arguments are ignored and the **Settings Window** will be opened result: value corresponding to ITEM

1.84 getarexxprefs

GetARexxPrefs: syntax: GetARexxPrefs ITEM/A function: Gets the current ARexx settings input: ITEM - one of the following values: F1,F2,...,F10 - name of the ARexx script on the function key ONLY_NAMES - returns YES or NO result: value corresponding to ITEM

1.85 setarexxprefs

SetARexxPrefs: syntax: SetARexxPrefs F1/K,F2/K,F3/K,F4/K,F5/K,F6/K,F7/K,F8/K,F9/K,F10/K, ONLY_NAMES/K,PROMPT/S function: Sets the current ARexx settings input: F1,F2,...,F10 - name of the ARexx script for the function key ONLY_NAMES - returns YES or NO PROMPT - If this argument is specified, all other arguments are ignored and the **Settings Window** will be opened result: value corresponding to ITEM

1.86 getexportprefs

GetExportPrefs: syntax: GetExportPrefs ITEM/A function: Gets the current export settings input: ITEM - one of the following values: PS_FONT - Include font into EPS file? Returns YES or NO BINEPS_FONT - Include font into Binary EPS file? Returns YES or NO BINEPS_DPI - Returns x and y dpi of the preview seperated by a 'x' IFF_DPI - Returns x and y dpi seperated by a 'x' TIFF_DPI - Returns x and y dpi seperated by a 'x' result: value corresponding to ITEM

1.87 setexportprefs

SetExportPrefs: syntax: SetExportPrefs PS_FONT/K,BINEPS_FONT/K,BINEPS_DPI/K,IFF_DPI/K, TIFF_DPI/K,PROMPT/S function: Sets the current export settings input: PS_FONT - Include font into EPS file? YES or NO BINEPS_FONT - Include font into Binary EPS file? YES or NO BINEPS_DPI - x and y dpi of the preview seperated by a 'x' IFF_DPI - x and y dpi seperated by a 'x' TIFF_DPI - x and y dpi seperated by a 'x' PROMPT - If this argument is specified, all other arguments are ignored and the **Settings Window** will be opened result: value corresponding to ITEM

1.88 SavePrefs

SavePrefs: syntax: SavePrefs function: Saves the current settings. input: - result: -

1.89 SavePrefsAs

SavePrefsAs: syntax: SavePrefsAs FILE,FORCE/S function: Saves the current settings into any file. If File is not specified, a filerequester will be opened input: - FILE - Name of the file FORCE - surpresses any warnings result: -

1.90 LoadPrefs

LoadPrefs: syntax: LoadPrefs FILE function: Loads a prefs file. If FILE is not specified, a filerequester will be opened. input: FILE - name of the file result: -

1.91 previewprefs

Preview: syntax: Preview function: Shows the preview of the current file input: - result: -

1.92 GetDimensions

GetDimensions: syntax: GetDimensions function: Returns the size of the formula input: - result: - size in format "<width><height>"

1.93 SetZoom

SetZoom: syntax: SetZoom ZOOM/A/N function: Set the zoom level input: ZOOM - number between 1 and 999 (in percent) result: -

1.94 GetZoom

GetZoom: syntax: GetZoom function: Get the zoom level input: - result: -

1.95 codes

Overview of all control codes

Up: syntax: $\langle \text{Obj1} \rangle^{\langle \text{Obj2} \rangle}$ function: Moves $\langle \text{Obj2} \rangle$ dependent on the size of $\langle \text{Obj1} \rangle$ up for example in order to show it as an exponent of a power. input: $\langle \text{Obj1} \rangle$ - base $\langle \text{Obj2} \rangle$ - exponent

Down: syntax: <Obj1>_<Obj2> function: Moves <Obj2> dependent on the size of <Obj1> down for example in order to show an subscript. input: <Obj1> - base <Obj2> - subscript

Up and down: syntax: $\langle \text{Obj1} \rangle^{\langle \text{Obj2} \rangle} \langle \text{Obj3} \rangle$ function: Moves $\langle \text{Obj2} \rangle$ dependent on the size of $\langle \text{Obj1} \rangle$ up and $\langle \text{Obj3} \rangle$ down. input: $\langle \text{Obj1} \rangle$ - base $\langle \text{Obj2} \rangle$ - exponent $\langle \text{Obj3} \rangle$ - subscript

Above: syntax: $\langle \text{Obj} \rangle \uparrow \langle \text{Obj2} \rangle$ function: Moves $\langle \text{Obj2} \rangle$ above $\langle \text{Obj1} \rangle$. input: $\langle \text{Obj1} \rangle$ - base object $\langle \text{Obj2} \rangle$ - higher object

Example: $-1 \in \text{Fup}[-1] \rightarrow F$ (inverse function)

Under: syntax: <Obj1>\dn<Obj2> function: Moves <Obj2> under <Obj1>. input: <Obj1> - base object <Obj2> - lower object

Example: $\lim_{n \rightarrow \infty} [\lim_{n \rightarrow \infty} \frac{1}{n}] = 0$

Above and under: syntax: <Obj1>ud<Obj2><Obj3> function: Moves <Obj2> above and <Obj3> under <Obj1>. input: <Obj1>
- base object <Obj2> - higher object <Obj3> - lower object

Brackets:

syntax: input output (<Obj1> -> (<Obj1> {<Obj1> -> {<Obj1> \l[<Obj1> -> [<Obj1> \l[<Obj1> ->]<Obj1> \l[<Obj1> -> \<Obj1> <Obj1>)} -> <Obj1>)} <Obj1>|r] -> <Obj1>| <Obj1>|r[-> <Obj1>[<Obj1>|r[-> <Obj1>| <Obj1>| More-over there are special pointed brackets which are only available through the **selection gadgets**: <<Obj1> -> <<Obj1> <Obj1>> -> <Obj1>>>

function: Scales the brackets dependent on the size of <Obj1> input: <Obj1> - specifies the size of the brackets

Example: $\vdash 1 \mid ([1/2x] \mid r \mid \rightarrow \mid -x \mid \mid 2 \mid$

Fraction: syntax: <Obj1>/<Obj2> function: Creates a fraction. input: <Obj1> - nominator <Obj2> - denominator

Square root: syntax: `\sqrt{Obj1}` function: Creates a square root. input: `<Obj1>`

Root: syntax: \rt<Obj1><Obj2> function: Creates a root. input: <Obj1> - root exponent <Obj2>

Matrix: syntax: \mx<m><n><Obj1><Obj2>...<Objm*n> function: Creates a (m,n)-matrix. input: <m> - one-digit number of rows (m=1..9) <n> - one-digit number of columns (n=1..9) <Obj1>,<Obj2>,...,<Objm*n> - Elements of the matrix (row by row) example:

$$(\sqrt[3]{2} \sqrt[4]{1/2} \sqrt[4]{2+3} \sqrt{2} 0)$$
$$/1\backslash/1-\backslash|2|||4\,2,3|\backslash\quad /\backslash Y2\,0/$$

Underline: syntax: \ul<Obj1> function: Underline the object. input: <Obj1> - underlined object.

Overline: syntax: \ol<Obj1> function: Overline the object. input: <Obj1> - overlined object. example:

```
-- \ol[PQ] -> PQ
```

Because the characters '^','_','\` and '/' are used as control codes you can use them as normal characters by writing a single backslash ('\') in front of them. Thus: \`-> ^ _ -> _ \\\ -> \ \/ -> /`

1.96 Disclaimer

MathScript is shareware. It may be distributed and copied as long as the following conditions are fulfilled:

- The sales price must not be higher than the cost of an empty disk plus a nominal copying fee plus costs for shipping.
- All parts of the program and the documentation must be complete. The distribution of single parts is not allowed.
- MathScript or parts of it may not be sold together with commercial software without the written permission of the author.
- All parts of the software package must not be changed in any way.
- The keyfile of a registered user may not be distributed.
- The Author is not responsible for misuse or damage caused by the program.

The program may be used for 14 days. After that period the **registration** is obligatory for further usage.

post.library © Copyright Adrian Aylward

1.97 Registration

If you register for MathScript you will obtain the latest version of it and a keyfile. With this keyfile you are able and allowed to use future releases of MathScript. Moreover these annoying requesters will disappear.

The shareware fee is 15\$ or 20DM.

To become a registered user print out the **registration form** (or write a similar form) and fill it out.

Send it to:

Simon Ihmig Beim Rauhen Hause 30 22111 Hamburg Germany

1.98 Registration form

I want to register for the program MathScript V2.1. I will get a keyfile which allows me to use the program regularly. I guarantee that it is only for my personal use and that I do not distribute it.

Kind of payment:

☐ The fee of 15\$ or 20DM is enclosed

☐ The fee of 15\$ or 20DM is transfered to the following account: No: 1077/782033 BLZ: 200 505 50 (Germany) Hamburger Sparkasse Simon Ihmig

My address:

First Name: _____

Last name: _____

Street: _____

City: _____

Phone: _____

Country: _____

e-mail: _____

Date: _____ signature: _____

1.99 Author

Author:

Simon Ihmig

Address:

Beim Rauhen Hause 30 22111 Hamburg Germany e-mail: Ihmig@tu-harburg.d400.de

If you have any comments, questions or bug-reports, please write to me! E-mail preferred!

1.100 Future

There were several points left from the old 2.0 version. However these were pretty difficult to implement with the current "code" concept. So I decided to rewrite 40% of the program from scratch for the 3.0 version. It will then feature full WYSIWYG!

From where can I get new Updates?

All updates are posted to the Aminet (misc/math). Moreover there is always the last version available in this fine BBS:

NotreDame Login: "Gast" Directory: <---Support--->/MathScript

Phone numbers: (area code +49-(0)40)

2800074 - ZyXEL EG 19200 bps (Ringdown with 4 Ports!) 247816 - ACTIVE 28800 bps (Ringdown with 4 Ports!) 28050032, 28050033 - ISDN X.75

1.101 credits

Thanks go to:

- Stefan Stuntz for his MagicUserInterface **MUI**
- Adrian Aylward for his post.library
- Kai Nickel for the MathScript support of his first-class program "Graph2D"
- David Ekholm for his swedish translation - Osmo Ahvenlampi for his finnish translation - Christian Hoj for his danish translation
- and last but not least all registered users!

1.102 mui

This application uses

MUI - MagicUserInterface

(c) Copyright 1993/94 by Stefan Stuntz

MUI is a system to generate and maintain graphical user interfaces. With the aid of a preferences program, the user of an application has the ability to customize the outfit according to his personal taste.

MUI is distributed as shareware. To obtain a complete package containing lots of examples and more information about registration please look for a file called "muiXXusr.lha" (XX means the latest version number) on your local bulletin boards or on public domain disks.

If you want to register directly, feel free to send

DM 30.- or US\$ 20.-

to

Stefan Stuntz Eduard-Spranger-Straße 7 80935 München GERMANY
