MathScript

Simon Ihmig

COLLABORATORS

	TITLE : MathScript		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Simon Ihmig	August 26, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

MathScript

1.1 MathScript Documentation

```
                          MathScript

            The formula editor for Amiga computers

                         Version 3.0
               © Copyright 1994-96 Simon Ihmig

                        Documentation
```

1.2 Description

Description

This program is a mathematical equation editor. Like a word
processor that lets you write text, MathScript is used to write
mathematical formulas.
It offers you a variety of mathematical and physical symbols and lots
of objects like fractions, roots, brackets and matrices to compose
formulas, which can then be imported into word processors or DTP
applications to produce scientific documents.

Here is an overview of the features of MathScript V3.0:

* WYSIWYG (What you see is what you get)
  While you edit your formula it is displayed in its final look.
  Editing is done like a word processor, use the keyboard to enter
  characters, move the cursor with the cursor keys or the mouse, mark
  blocks by dragging the mouse...

* MUI
  Uses MUI 3.x for its nice font-sensitive resizeable and customizable
  GUI.

* Symbols, Objects
  Extensive amount of symbols and objects, accessable with the
  keyboard and/or popup menus

* PostScript Fonts
  Uses industry-standard scalable PostScript Type1 or Type3 fonts. You
  can choose different fonts for the following types of characters:
  variable, function, text, greek, symbol, number and 2 special
  purpose fonts.

* AutoFormat
  MathScript automatically formats your input, i.e. it uses the
  appropriate font for each character.
  So called "styles" affect the way this is being done. The following
  styles exist:
  Math, Variable, Function, Greek, Text, Extra1, Extra2.

* Function Recognition
  MathScript "knows" the most common function names (like sin) and is

```
able to automatically apply the function font to it
```

* Export
  Save formulas as EPS (PostScript), IFF or TIFF files

* Import
  Files produced by the export function can be loaded back again

* ARexx
  It has an ARexx port with lots of commands

* Clipboard
  Support the system clipboard for cut, copy and paste operations

* Function Keys
  ARexx scripts can be assigned to function keys

* Online Help
  It offers extensive online help in four different ways:
  – Menu Help: AmigaGuide help for menu items
  – Gadget Help: AmigaGuide help for gadgets
  – Bubble Help: Help bubbles for gadgets
  – Quick Help: Status bar help for gadgets and popup menu items

* Toolbar
  Access the most used functions with nice little image buttons

* Locale
  Supports locale.library (OS 2.1+) for localized GUI text

* Registration
  Registering is easily done with the little Registration Tool

1.3 System Requirements

System Requirements

To be able to run MathScript you must have at least the following
system:

```
  Amiga with 68000 processor
  2 MB RAM
  AmigaOS 2.0
```

For reasonable use of the program I recommend the following system:

```
  Amiga with 68030 processor
  6 MB RAM
  AmigaOS 3.x
  HardDisk
```

A WYSIWYG equation editor needs by its nature many system resources.
Therefore you will get the best results with a faster processor

(040/060) and/or a graphics card.

Moreover the following system extensions are needed:

post.library V17 (included)

MUI 3.x (I recommend to always look for the latest version!)

1.4 Starting

Starting

MathScript can either be started from the Workbench or from the Shell.

To start it from the Workbench, open the drawer where MathScript has
been installed to and double click on the "MathScript" icon.
You can also double click on a formula file which will launch
MathScript and load the formula automatically.

To start it from the Shell, go to the directory where the program has
been installed to (e.g. "cd Work:MathScript") and type "MathScript".
You can also specify an optional filename of a MathScript formula,
which will be loaded then.

After you have started the program in one of these ways, a status
window will appear while all resources (fonts) needed by the program
are loaded. The window will be closed automatically afterwards.

1.5 User Interface

User Interface

The Project Window is the main window where you enter your formulas
and access all the other functions of MathScript.

It will look like this (varying with your MUI configuration):

  <- click here (OS3.x only!)

Edit Area

The main part of the window is the edit area. Here you edit your
formulas.

Toolbar

The Toolbar consists of several image buttons that contain the most
often used functions of the program like saving, exporting and the
clipboard operations.

Style Selector

With the Style Selector you can chose the active style, which
determines how MathScript formats the formula.

Symbol Menus

With the 14 Symbol Menus you can select all symbols that you cannot
type with the keyboard as well as all those objects like fractions or
roots.

Status Bar

The Status Bar is used for MathScript's QuickHelp feature. It will
display small help texts as you move the mouse pointer over any button
or popup menu item.

Menu

From the menu you can access most of the program's features.

1.6 Toolbar

Toolbar

The Toolbar holds several buttons which start the following functions:
(from left to right)

 New Project
 Open Project
 Save Project

 Export
 Import

 Cut
 Copy
 Paste

 Execute ARexx script

 Change Settings

 Zoom

1.7 Zoom

Zoom

Use this button to change the magnification of the formula display.
The values you adjust here in % are relative to the display resolution

settings.

1.8 Style Selector

Style Selector

The current style will be displayed here. Use the
popup button to the right to change it.

1.9 Online Help

Online Help

MathScript has a very extensive Online Help system.
It features four kinds of help:

Menu Help

If you browse through any menu and do not know what a certain menu
item is doing, simpley press the Help key while this item is active.
An AmigaGuide node will be displayed which explains the menu item.

Gadget Help

If you point on a gadget with the mouse and press the Help key, an
AmigaGuide window will be opened that explains the function of the
gadget.

Bubble Help

This MUI feature is invoked by letting the mouse pointer rest over a
button for a specific amount of time (configurable in the MUI
settings). A Help bubble will then appear that contains some short
information on the gadget.

Quick Help

While you move your mouse pointer over the Project Window, the Status
Bar will display a very short help text for the area the mouse is
currently pointing to. This also works for all the popup menu items.
It is especially useful for the Symbol Menu items when you do not
know what a certain symbol is standing for.
Note that this kind of help is only available in the Project Window!

1.10 Editing

Editing

This chapter describes how to edit formulas.

The way it is done is very straightforward. If you are familiar with
WYSIWYG word processors, you will find it very easy, so probably you
can skip this chapter.

Insert Text

To insert characters in MathScript, there are two differnd ways to do
this. You can either use your keyboard or the Symbol Menus for special
symbols. In every case the symbols are inserted at the current cursor
position (displayed as a vertical bar).

With the keyboard

Use the keyboard to insert regular alphabetic characters, numbers and
some common symbols like '+' or '='.
MathScript supports the full Amiga character set (ISO Latin), so you
can even use specific national characters like french vocals with
accents (e.g. á) or german "umlauts" (e.g. ä).

With the Symbol Menus

The Symbol Menus are used to insert special mathematical symbols and
objects.

Move the cursor

Just use the cursor keys to move it, by holding down the shift key it
will jump to the beginning/end of the line.
You can also use the mouse to position it, simply point to the place
where the cursor should be and press the left mouse button.

Mark blocks

For certain operations it is needed to have a block of your formula
marked. To do this, move the mouse to one end of the block, press the
left mouse button and move the mouse to the other end. Then release
the mouse button again. The part of the formula you have selected will
be displayed inverted.

Note: if you have a block marked and you insert a symbol, the block
will be replaced by the new symbol. If you insert an object, it can
happen (depending on the object) that the block will be put as a part
of the object. For example if you have marked a block and insert a
bracket object, the block will be placed inside of the brackets.

Delete symbols

Use either the DEL or the Backspace key to delete parts of your
formula. If you have marked a block, the whole block will be removed
if you press one of the keys. Otherwise the symbol to the left or the
right will be deleted, depending on which key you have pressed.

The Clipboard

The clipboard lets you cut, copy and paste formula parts.
If you have marked a block and cut it, it will be removed and placed
into the clipboard. Later you can paste it back into your formula at
the current cursor position. The copy function will also put the block
into the clipboard, but will not remove it. These functions are to be
used if you want to move or duplicate parts of your formula.

They can be accessed in three different ways:
a) select one of the menu items in the Edit menu,
b) use the common keyboard shortcuts (displayed in the Edit menu) or
c) press on of the clipboard buttons in the Toolbar.

1.11 Formatting

Formatting

Formatting text usually means choosing typefaces, adjusting font
sizes, setting paragraph justification and so on.
However this is not the way formatting formulas is done with
MathScript.

Mathematical formulas follow certain rules like using regular (plain)
typefaces for functions and italic ones for variables, or decreasing
the font size for exponents. There is no creative or artistic element
in layouting formulas.

Therefore MathScript tries to do this almost automatically.
It tries to find the proper font for every character, adjusts the
space between them automatically and decreases the font's size for
certain parts liek exponents or fractions.

Read on to see how MathScript does this!

FunctionRecognition

MathScript has a special feature called FunctionRecognition.
While you edit your formula, MathScript will scan the changed area for
strings that are known as standard function names like "sin" or "log".
If it finds one, it is able to automatically assign the function font
to it.

SpaceAdjustment

Many mathematical symbols (like +,-,=,<,...) will have their space to
the next/previous character automatically increased.

Character Types

MathScript "knows" the following types of characters:
1. variables

2. function names
3. text characters
4. greek characters
5. (mathematical) symbols
6. numbers

Each type of character can have its own font. Moreover you can use two
special fonts for characters not available elsewhere.

Now you do not have to specify for each character what type it should
be. MathScript distinguishes the following types of characters
automatically:

– Alphabetic characters. Everything that's not one of the following
  types.
– Numbers. All characters from '0' – '9'.
– Symbols like '+' or '<'.
  (Note that symbols that you insert with the symbol menus will not be
  affected. Since they are not available in regular roman fonts, they
  will always use the symbol font.)

However this is not enough to automatically chose the right font for
every character.

For example MathScript has to know if alphabetic characters should be
roman or greek.
And for things like comments in your formulas it is not useful to have
numbers use a different font than alphabetic characters.

For this reasons MathScript uses so called "Styles".

Styles

A style defines how MathScript will choose the character type and thus
the font to be used for alphabetic characters, numbers and symbols.

The active style can be changed with the Style Selector.

All characters that you enter with the keyboard will use the current
style. The style of existing parts of a formula can be changed by
marking a block and selecting a new style.

A list of all available styles can be found here.

1.12   List of all styles

List of all styles

For each style there is a comment that describes when to use them.
It also tells you which type of character recognized by MathScript
will use which font:

 Math

```
 This is the standard style.
 Everything (almost) will be formatted as it should be for regular
 mathematical formulas.
 It uses FunctionRecognition to seperate regular alphabetic
 characters from those that are part of a function name.
 Characters recognized as symbols or numbers will use the symbol or
 number font respectively.

 Variable

 The same as Math except that it does not use FunctionRecognition.
 Instead it always uses the variable font for alphabetic characters.
 Usually you won't need this style.

 Function

 The same as Math except that it does not use FunctionRecognition.
 Instead it always uses the function font for alphabetic characters.
 Apply this style to function names that are not recognized as such
 because they are not defined as a standard name (like f(x) ).

 Greek

 The same as Math except that it does not use FunctionRecognition.
 Instead it always uses the greek font for alphabetic characters.
 You can use this style to type greek characters with the keyboard.
 For example entering an 'a' will insert an alpha if this style is
 active.

 Text

 The text style does not distinguish different types of characters.
 Every character will use the text font.
 Use this style for things like comments.

 Extra1
 Extra2

 The Extra styles do not distinguish different types of characters.
 Every character will use the Extra1 or Extra2 font respectively.
 If you have fonts with special characters not available elsewhere,
 use these styles to access them.
```

1.13   The Symbol Menus

```
The Symbol Menus

In the main window are 14 buttons which let you insert a lot of
symbols and objects that are not accessable with the keyboard.

When you click on one of them, a menu will pop up under it.
It works like a regular menu, i.e. keep the (left) mouse button
pressed, move the mouse pointer and release the button to select a
```

menu item.

When you have selected one item, the symbol or object it represents
will be inserted at the current cursor position.

Note:
The QuickHelp feature is very handy for the symbol menus. If you move
the mouse pointer over a symbol button, the name of the group of
symbols will be displayed (as listed below) in the status bar. If a
menu item is highlighted, the name of the symbol/object will be shown.
For some menu item there are also keyboard shortcuts. If there is one,
it will be shown in brackets in the QuickHelp text.
For example you can type <control>-'a' to insert a lowercase alpha.

There are the following symbol buttons:

First Row:

1. Operators

Mathematical operators for addition, multiplication, ...

2. Arrows

Single and double arrows with different directions.

3. Logical symbols

Logical symbols like AND, OR, NOT, ...

4. Brackets

Brackets, paranthesis, braces and vertical bars.

5. Product, Summation, Integral

Greek Symbols for Products, Summations and Integrals.

6. Lowercase greek symbols

All lowercase greek symbols sorted alphabetically (refering to the
greek alphabet). For some characters there are two different versions,
so both are available.

7. Uppercase greek symbols

All uppercase greek symbols sorted alphabetically (refering to the
greek alphabet).

Second Row:

8. Relational symbols

Symbols for equations and relations.

9. Set theory symbols

Symbols from the theory of sets like union or subset.

10. Miscellaneous symbols

Symbols not belonging to other groups.

11. Superscript, subscript

Objects to super/subscript symbols or place them above/under others.

12. Fraction, root

Objects for fractions and (square-)roots.

13. Derivatives, vectors, underline

Objects that implement derivatives (x or t), vectors (arrows above) or
under/overlines.

14. Matrices

Matrices with different numbers of rows and columns. Note that the
last menu item does not insert anything at once but opens the
Matrix Window where you can adjust the numbers of rows/columns.

1.14 Matrix Window

Matrix Window

The Matrix window has two sliders which can be used to specify the
number of rows and columns for a matrix you want to insert.

1.15 Saving and Exporting

Saving and Exporting

Saving

If you want to save your formula to disk, use the menu item
Project/Save as. A file requester will appear where you can enter the
name of your file. If you have already saved your formula, you can
also use the menu item Project/Save. It will not ask you for a file
name but use the current file name instead.

Loading

If you have saved a formula in a way described above, you can use the
menu item Project/Open to load it. A file requester will be opened
where you have to choose the file you want to load.

Export

Files that you create by exporting a formula use a common file format
that can be read by other programs. Use the menu item Project/Export
for this. Its sub menu will let you choose between the different
file formats MathScript is able to create.

Import

A file created by the Export function of MathScript can be loaded back
by using the menu item Project/Import.

1.16 File Formats

File Formats

EPS

EPS means "Encapsulated PostScript".

PostScript is the industry standard page description language.
It is a powerful programming language designed for describing the
appearance of text and graphics on a printed page or display.
A PostScript interpreter handles the task of converting the language
code to raster images. It can either be a software run on a computer
or part of an output device like a laser printer.

The EPS files generated by MathScript can have a TIFF bitmap preview
included for fast on-screen display.

IFF

The IFF-ILBM format is the standard on the Amiga for bitmap graphics.
Probably all Amiga software that is able to handle bitmap graphics
will understand this format.

Because it is a bitmap format you will not reach the printing quality of
EPS files.

TIFF

The TIFF format is a very popular file format for bitmap graphics,
especially on other computer platforms.

1.17 Settings

Settings

The user adjustable settings for MathScript can be changed with the
Settings Window. You can open it with the Settings/Programm... menu
item or the appropriate Toolbar button.

It is divided into the following five different "pages":

System

General

Fonts

ARexx

Export

Moreover it has an own menu with the following items:
 Reset to Defaults
 Last Saved
 Restore

If you have changed any settings and wish use them, leave the Settings
Window by clicking in the "Ok" button. Otherwise click on "Cancel" to
ignore any changes.

1.18 System Settings

System Settings"

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

Path

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

Create Icons

If this option is active MathScript will create suitable icons for
saved/exported files.

1.19 General

General

On this page you can set several general preferences.

Font Size

You are able to specify a font size in pt (1/72 inch) here.
The size you specify here is used for all parts of your formula at the
highest "level". Symbols that are part of some objects will
automatically use a smaller size, derived from this setting.

Display Resolution

Both gadgets can be used to set the x and y resolution of the formula
screen display in dpi (dots per inch).
You can specify the dpi resolution of your monitor (pixels/inch) here
to have real WYSIWYG, i.e. the size of the screen display will be
equal to your final output. However it may be more useful to specify a
larger resolution for easy editing.

1.20 Fonts Settings

Fonts Settings

MathScript lets you choose different fonts for certain types of
formula parts.
You can use any PostScript Type1 or Type3 fonts as long as each of
them has an AFM (Adobe Font Metrics) file. The name of an AFM file
must be the name of the font file (without the suffix ".pfb"/".pfa")
with the suffix ".afm" appended.

You can choose a font for the following types:

Variable

This font will be used for variables, i.e. characters that are not
part of a function name or mathematical symbols.

Function

Function names will be written using this font.

Text

If you want to include comments or other regular texts that are not to
be formatted like a formula with the Text style, all characters will
use this font.

Greek

This font will be used for all greek characters.
Note that you should not use a regular roman font here, but a font
that uses SymbolEncoding like the original "Symbol" font.

Symbol

All mathematical symbols that you insert with the Symbol Menus will be

shown using this font.
Note that you should not use a regular roman font here, but a font
that uses SymbolEncoding like the original "Symbol" font.

Number

This font will be automatically used for numeric characters.

Extra1
Extra2

These two optional fonts can be used for special purposes.
For example you can insert special symbols that are not available
otherwise when you have special mathematical extensions fonts.

For recommendations of fonts to be used see the FAQ section.

1.21   ARexx Settings

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.
Pressing the PopUp gadgets on the right side will open an file
requester.

1.22   Export Settings

Export Settings

You can set some options for the different graphics file formats on
this page.
It consists of some other pages which contain settings for a specific
format, namely:
EPS
IFF
TIFF

EPS Settings

Include Fonts

If this option is activated the PostScript fonts in use will be
embedded into your EPS files.
This has the advantage that your files will be independand of

available resources (fonts) your PostScript interpreter has access to,
but your files will grow considerably.
Note that only those fonts will be included that are not listed in
your Files/FontList file.
For more information have a look at the FAQ!

Use Preview

EPS files may have a preview bitmap included as a TIFF file. This can
be used for fast screen display with your word processor.
Use this option if your importing application is able to take
advantage on it.

Preview Resolution

If you have activated the "Use Preview" option, these gadgets can be
used to specify the resolution in dpi of the TIFF preview.

IFF Settings

Resolution

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

TIFF Settings

Resolution

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

1.23   Reset to Defaults

Reset to Defaults

If you select this menu item the settings will be set back to the
defaults.

1.24   Last Saved

Last Saved

The settings of the default settings file will be loaded again.

1.25   Restore

Restore

The settings will be set to the state they had before opening the

Settings Window.

1.26   Other applications

Other applications

The formulas produced by MathScript will usually be used as parts of
larger documents created by word processors or desktop publishing
programs.

The general method to produce such documents is like this:

1. Enter the formula with MathScript
2. Export it as a graphics file
3. Import this file into your application

Moreover there are some special instructions for the following Amiga
programs:

FinalWriter

WordWorth

1.27   FinalWriter

FinalWriter

MathScript comes with some ARexx scripts that do the boring job of
exchanging formulas between MathScript and FinalWriter almost
automatically.
Before you can use them, some installations have to be done.

Setup of...

 init.ps

  FinalWriter uses its built-in PostScript interpreter for printing
  and displaying EPS files. In order to be shown correctly FinalWriter
  has to find the fonts used by these EPS files. Therefore some
  modifications have to be made with the "init.ps" file in the FWFiles
  directory. If these modifications have already been made during the
  installation of MathScript, you can continue reading. Otherwise have
  a look at Change init.ps now!

 Public Screen

  Since version 5 FinalWriter uses a public screen. This comes very
  handy since it allows MathScript's windows to be opened on the
  FinalWriter screen. To make this happen, you have to tell MUI which
  screen to use.

Select the menu item Settings/MUI, go to the "System" page and
select FinalWriter's screen from the screen popup button. If you
have not FinalWriter running, enter "FinalWriterPubScreen".
As an alternative you can also select "Frontmost", which
obviously uses the screen at the front.

 ARexx scripts

  There are two ARexx scripts that are used very often, so you should
  assign them to a menu item/function key/user button.
  In MathScript you should install the ARexx script
  "FW_InsertFormula.ms" to a function key/menu item of your choice.
  In FinalWriter you should create a user menu item or toolbar button
  executing the ARexx script "FW_OpenMathScript.ms".
  (These scripts are located in the MathScript:Rexx directory)

Usage

 If you have followed the setup instructions above, you are ready to
 use the supplied FinalWriter ARexx scripts, which are:

 FW_OpenMathScript.ms
 FW_InsertFormula.ms
 FW_Cleanup.ms

 Whenever you want to create a new formula, start the
 FW_OpenMathScript.ms script. This will start MathScript or uniconify
 it if it is already running. The current font size will be set to the
 current size of FinalWriter making your formulas equally large as
 your FinalWriter text. (you can change this within the script)
 Now you can enter your formula as usual.

 When the formula is complete, execute the FW_InsertFormula.ms script.
 This will generate a temporary EPS file, which is automatically
 imported into FinalWriter

 When you do not want to create a new formula but edit an existing
 one, just select the formula and continue with the proceeding
 described above. MathScript will import the formula so you can edit
 it. When you insert the modified formula with the FW_InsertFormula.ms
 script, the old formula will be replaced by the new one in
 FinalWriter.

 When you have finished your work and quited FinalWriter (!) you
 should execute the FW_Cleanup.ms script that deletes temporary EPS
 files from T:.

1.28 WordWorth

WordWorth

Unfortunately the ARexx port of WordWorth V5 is not complete enough to
support automated exchange of formulas with MathScript. Nevertheless

you should be able to use both programs without any problems,
exporting and importing files manually.

Setup

 Wordworth's built-in PostScript interpreter has to find the fonts
 used by MathScript, so you have to make some modifications of the
 eps_init.ps file in the WWFiles directory. If this has not been done
 already during the installation of MathScript, do it now!
 See under Change init.ps for instructions.

Usage

 When you have entered a formula in MathScript, you should export it
 as an EPS file. In WordWorth select the "Place picture" (or
 equivalent) menu item to import this EPS file. If the setup has been
 correctly done, you should see the formula on the screen.

1.29 Change init.ps

Change init.ps

Load the file into an editor and find the place where the findfont
operator is defined. You have to make it look for fonts in the
PSFonts: directory with the possible extension of .pfb.
Replace the findfont section with this one:

```
--- CUT HERE ---
systemdict
/findfont
  {
  dup FontDirectory exch known not                 % Not in FontDirectory?
  { dup (PSFonts:) () loadfont                      % Try to load from PSFonts:
  } if
  dup FontDirectory exch known not                 % Not in FontDirectory?
  { dup (PSFonts:) (.pfb) loadfont                  % Try to load from PSFonts:
  } if

% Substitute DefaultFont
  dup FontDirectory exch known not                 % Not in FontDirectory?
  { dup userdict /DefaultFontName get ne            % Not the DefaultFont
    { pop userdict /DefaultFontName get             % Substitue DefaultFont
      dup systemdict /findfont get exec pop          % Recurse so it gets loaded
    }
    if
  }
  if
  dup FontDirectory exch known not                 % Not in FontDirectory?
  { pop /DummyFont } if                             % Fall back on DummyFont
  .findfont                                          % If errors, will fail now
} bind put
--- CUT HERE ---
```

Note that a % begins a comment, you can ignore them.

1.30 Menu

Menu

The Menu of the Project Window has the following items:

Project

New
Open...
Save
Save as...
Import...
Export
 EPS
 IFF
 TIFF
About...
Quit

Edit

Cut
Copy
Paste

ARexx

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

Settings

Program...
MUI...
Load Settings...
Save Settings
Save Settings as...

1.31 New

New

There current formula will be cleared and the current file name will
be set back to "untitled".

1.32 Open

Open

This menu item opens a file requester in which a file can be chosen
which should be loaded.
Only files which have the suffix .fml are shown here. 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 Project Window or AppIcon (if iconified).

1.33 Save

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
name will be used for saving.

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

1.34 Save as

Save as

This saves the current formula to a file in order to reuse or modify
the formula later on. A file requester will be opened to select a file
for saving.

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

1.35 Import

Import

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

If an exported file was modified in any way by another program, the
whole file will be searched for the formula description.
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.

1.36   Export

Export

This function saves the current formula as a graphics file which can be
loaded into other applications like word processors or DTP
applications.

MathScript supports different file formats which are selectable as
submenu items of the Project/Export menu item.

1.37   About

About

The About Window will be opened. It contains information on MathScript
such as the current version number or the name of the ARexx port.

The Register button will load the Registration Tool, allowing you to
get the registered version of MathScript.

The "About MUI" button will show the information window of MUI.

1.38   Quit

Quit

Quit the program.

1.39   Cut

Cut

If there is a marked formula block it will be cut, i.e. the block will
be removed from the formula and placed into the system clipboard.
You can later insert it with the Paste command.

1.40   Copy

Copy

If there is a marked formula block it will be copied to the system
clipboard.
You can then insert it with the Paste command.

1.41 Paste

Paste

The contents of the system clipboard will be inserted at the current
cursor position.
Note that real formulas will only be inserted if they were placed into
the clipboard by MathScript itself. Otherwise the contents of the
clipboard will be inserted as regular text.

1.42 Execute ARexx script

Execute ARexx script

A file requester will be opened to choose an ARexx script that should
be executed.
Look under Arexx for detailled informations on MathScript's ARexx
capabilities.

1.43 Function Keys

Function Keys

If you have installed an ARexx script to a particular function key,
you are also able to execute that script by selecting the
corresponding menu item.
Look under Settings to learn how to assign ARexx scripts to
function keys.

1.44 Program Settings

Program Settings

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

1.45 MUI Settings

MUI Settings

This menu item will bring up the MUI Preferences Window. You can
change all MUI settings here which will only affect MathScript.

1.46 Load settings

Load settings

You can load the settings from a previously saved settings file with
this menu item.

1.47 Save Settings

Save Settings

This will save your default settings which will be used next time you
start MathScript.

1.48 Save Settings as

Save Settings as

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

1.49 ARexx Preface

ARexx Preface

ARexx is the Amiga's version of the Rexx programming language. It is
used for inter-process-communication and automation.
The programs written in ARexx are called ARexx scripts.
For more information on ARexx see the ARexx documentation that came
with your Amiga.

MathScript has an extensive ARexx port with lots of commands.
You can use it for macros and to automate the exchange of formulas
with word processors.

MathScript's ARexx port is called "MATHSCRIPT.1" for the first task,
"MATHSCRIPT.2" for the second and so on.

MathScript ARexx Commands

1.50 Formula Code

Formula Code

For saving formulas and for some ARexx commands MathScript uses
standard ASCII text to represent its formulas. Objects like fractions

or roots are written with special codes which are being described now.

Note: Some AmigaGuide versions handle backslashes differently than others. If this backslash \ is shown twice on your system, ignore this.

Characters

Regular characters are represented by a single ASCII character. You can specify the style of the character by one of the following control codes:

```
  Control Code    Style      Encoding

     \ma          Math       ISO Latin 1 (Amiga ASCII)
     \va          Variable   ISO Latin 1 (Amiga ASCII)
     \fu          Function   ISO Latin 1 (Amiga ASCII)
     \te          Text       ISO Latin 1 (Amiga ASCII)
     \sy          (Symbol)   Adobe SymbolEncoding
     \gr          Greek      Adobe SymbolEncoding
     \c1          Extra1     ISO Latin 1 (Amiga ASCII)
     \c2          Extra2     ISO Latin 1 (Amiga ASCII)
```

If you do not specify anything the Math Style will be used.

Example:

  \fug(x)=\gra

  will produce this formula: g(x)=<alpha>

Objects

If you want to make objects like fractions or roots use one of the following control codes listed below.
Note that objects have children objects which must be enclosed by brackets like this: [<characters and objects>].


```
Name           Control Code            Function

 Exponent      ^<obj>                  Make <obj> an exponent
 Index         _<obj>                  Make <obj> an index
 Exp&Index     ^_<obj1><obj2>          Make <obj1> an exponent and
                                       <obj2> an index

 Up            <obj1>\up<obj2>         Place <obj2> above <obj1>
 Down          <obj1>\dn>obj2>         Place <obj2> below <obj1>
 Up&Down       <obj1>\ud<obj2><obj3>   Place <obj2> above and <obj3>
                                       below <obj1>

 Left Bracket  \l<b><obj>              Make a <b> bracket sized to the
                                       height of <obj>
 Right Bracket <obj>\r<b>              Make a <b> bracket sized to the
                                       height of <obj>
```

		<b> can be one of the following characters: ( ) [ ] { } < > |
Fraction	<obj1>/<obj2>	Make a fraction. <obj1> is the nominator and <obj2> the denominator
Root	\rt<obj1><obj2>	Make a root. <obj1> is the root exponent
Square root	\sqrt<obj1>	Make a square root.
Matrix	\mx<m><n><obj1>...	Make a matrix with <m> rows and <n> columns. <m> and <n> must be numbers betwenn 1 and 9. <m> * <n> objects have to follow.
Underline	\ul<obj>	Underline <obj>
Overline	\ol<obj>	Overline <obj>
Derivative	\dx<n> \dt<n><obj>	Make a x or t derivative. <n> = 1-3.
New line	\n	New line

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.

Examples

\l([\mx31[1][3/2][4]]\r)

\fuf\dx1(x)=sin\sqrt[x]

1.51 Legal Information

Legal Information

Disclaimer

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING

```
MathScript is (C) Copyright 1994-96 Simon Ihmig.

post.library is (C) Copyright Adrian Aylward.

The Utopia and Symbol fonts are (C) Copyright Adobe Systems Inc.

Amiga is a registered trademark of ESCOM AG.
Adobe, PostScript and Utopia are trademarks of Adobe Systems Inc.
Times is a trademark of Linotype AG and/or its subsidiaries
WordWorth is a registered trademark of Digita Holdings, Ltd.
FinalWriter is a trademark of SoftWood Inc.
Other brand or product names are the trademarks or registered
trademarks of their respective holders.
```

1.52 Registration

```
Registration

MathScript is shareware.
It means that you can test it for 21 days with the restriction that
you are not able to load/save your settings. After that period you
have to register the program.

You do this by providing me with your name and address and the
registration fee, of course! :-)
You will then receive your personal keyfile. This keyfile will allow
you to use the program regularly. Future minor updates will be without
cost for you.

To register simply use the supplied "Registration Tool". It is located
in the MathScript directory. You can start it from there, from the
reminder requester or the About Window or directly from here.
It has a Help window which will explain how to use it.
```

1.53 MathScript Support

```
MathScript Support

If you have any questions, suggestions, problems, bug reports or
whatever, please don't hesitate to contact me!

My Address:

  Simon Ihmig
  Beim Rauhen Hause 30
  22111 Hamburg
  Germany
```

```
  Fax:   +49 (0)40 65901168
  EMail: Ihmig@tu-harburg.d400.de
```

1.54 MathScript History

```
**************************
*** MathScript history ***
**************************


Version 1.0, Release Date: 19.11.94
-----------------------------------
  First release


Version 2.0, Release Date: 12.04.95
-----------------------------------
    - Needs MUI (Magic User Interface ©Stefan Stuntz). Therefore some changes
      were required:
      - Because MathScript is now able to have multiple copies running,
        the name of the ARexx port changed. Now it is: "MathScript.1" for the
        first task, "MathScript.2" for the second and so on...
      - The syntax of the ARexx commands has changed, some commands
        are not compatible with the previous release
      - New, more convenient Selection-Gadgets (MUI custom class)
      - No public screen settings, no PubScreen/K CLI parameter, use the
        MUI Prefs program instead
    - Offering Bitmap output facility. Thus the following two points were
      made possible:
      - Preview of formula
      - Added support for the following export formats:
        - Binary EPS (EPS + TIFF preview)
        - IFF-ILBM
        - TIFF
    - ARexx scripts on function keys and menu items
    - Settings are saved into a file instead of ToolTypes,
      new menu items Load/Save_as settings
    - EPS output conforms to EPSF 3.0 specifications
    - EPS code is 100% Ascii. That should fix some problems that could occur
      on old PostScript printers
    - Fixed bug which caused brackets on the right side to appear too small
      sometimes
    - Online Help (AmigaGuide)
    - Some new ARexx commands
    - AppWindow/AppIcon support
    - Double,triple,... fractions/exponents were too big/small. They look
      fine now
    - Formula description is included into export files. New menu item "Import"
      allows you to load a formula from an export file
    - The usage of the FinalWriter ARexx scripts has changed slightly.
      By using the Import feature you are able to edit formulas inserted into
      FinalWriter later
    - Some minor changes and improvements


Version 2.1, Release Date: 02.06.95
-----------------------------------
    - New control codes for:
```

```
      - Matrices
      - Over-/Underline
    - Toolbar icons (8 colors like MagicWB) for the most used menu items
    - Zoom of preview
    - Quick Help (Gadget Help). Put the mouse pointer over a gadget and a little
      help text wil be shown
    - adjustable Fore-/Background color for exported formulas
    - Installer script now also in German
    - New ARexx commands "SetZoom" and "GetZoom", GeneralPrefs ARexx commands
      support color
    - ARexx script InsertFormula.ms has been improved. Now it uses the current
      font color. (you may change this!)
    - removed a bug in a formula interpretation function which crashed on
      68000 systems
    - removed a stupid bug that caused a dramatical rendering slowdown
      of *very* nested formulas. Now MUCH faster!
    - bug which screwed up the preview after iconification fixed
    - Several minor bug fixes and improvements
```

```
Version 3.0, Release Date: 25.09.96 (Major Update)
----------------------------------
    The program has been totally rewritten, mostly from scratch.
    The main difference is that it now works in a WYSIWYG manner.
    No more control codes have to be used, editing is done directly
    in the formula display.
    Here is just a rough overview of other major changes:

    - MathScript now does not use a custom math font. Instead you can choose
      any PostScript font, means Type1 and even Type3 fonts. Corresponding
      AFM files are needed.

    - one can choose different fonts for the following types:
      Variable, function, text, greek, symbol, number and 2 special purpose fonts.

    - Styles: Parts of a formula can be formatted differently by applying
      so called styles, i.e. MathScript chooses the proper font for every part of
      the formula. For example the greek style will make MathScript use the greek
      font for alphabetic characters, the number font for numbers, ...
      The following styles are available:
      Math, Variable, Function, Greek, Text, Custom1, Custom2.

    - Function Recognition: Every part of a formula which is using the default
      Math style will be scanned for a number of known function names, which
      then will use the function font. E.g. typing "sinx" will make MathScript
      use the function font for "sin" and the variable font for "x".

    - Clipboard: the standard clipboard operations (cut, copy and paste) for
      marked blocks are implemented (using clipboard.device!)

    - Multiple rows: it is now possible to have formulas with many rows simply
      by pressing the return key.

    - Better quality: the formulas are looking better now, especially brackets
      and roots.

    - Advanced Online Help: MathScript has one of the most advanced online help
      systems, featuring GadgetHelp, MenuHelp, BubbleHelp and QuickHelp (for every
```

```
      popup menu item!)
```

 – MUI 3.x: MathScript will only run if at least MUI 3.1 is installed.
   MUI 3.3 is recommended! It uses its advanced features like bubble help.

 – Formula Code: the old control code is not needed anymore, but it is still
   supported for file and ARexx I/O, means old formula files are still valid.

 – ARexx: The ARexx port has been reworked. The Settings commands support
   ARexx variables now. The new interface is mostly incompatible with the old.

 – Registration: there is a new Registration Tool handling the process of
   registration.

 – Many other changes and improvements

1.55 Frequently Asked Questions

```
Frequently Asked Questions

I need a special symbol that is not available!

MathScript relies on external fonts for its set of symbols. If you
need a symbol that is not available as part of the Symbol font, you
can install special math fonts as your extra fonts.
See also Settings/Fonts and the next items.


What fonts can I use?

You can use every PostScript Type1 (or even Type3) font with
MathScript. For each font you need the font file itself (.pfb) and its
Adobe Font Metrics file (AFM). In order to be able to associate every
font file with its AFM file a common naming convention is used:
The name of the AFM file must be the name of the font file (without
the suffix .pfb or .pfa) with the suffix .afm.
For example if you have a font file named Times-Roman.pfb, the AFM
file must have the name Times-Roman.afm.
See also Settings/Fonts.


What fonts should I use?

MathScript comes with some font files that are freely available.
However for better quality you may want to replace them with other
(commercial) fonts. I suggest you to use the Symbol font with fonts
from the Times family.
This is my setup:

Variable: Times-Italic
Function: Times-Roman
Text:     Times-Roman
Greek:    Symbol
Symbol:   Symbol
Number:   Times-Roman
```

You may choose special math fonts which have symbols not contained in
the standard Symbol font for your extra fonts settings.
The "Mathematical Pi" package from Adobe includes six special math
fonts. While not very cheap it contains a huge amount of symbols.
The fonts mentioned here are available directly from Adobe Systems
Inc.

Note that I take no responsibility for these font suggestions.
They are totally subjective and do not represent in any way all
products suitable or available.

Some WWW pages of interest:
Adobe Systems Homepage http://www.adobe.com/
Adobe Type Browser   http://www.adobe.com/type/browser/main.html
Internet Font Archive   http://jasper.ora.com/compfont/ifa/ifa.htm

What if I do not have a PostScript printer

If you want to process PostScript (EPS) files you need a PostScript
interpreter. If you do not have one built into your printer you have
to use a software interpreter. Some applications like FinalWriter or
WordWorth have one built-in. If the application you use with
MathScript does not have one but is able to output PostScript files,
you can print this PostScript file with a seperate PostScript
interpreter. There are some freely available like:

GhostScript by Aladdin Enterprises
Post by Adrian Aylward
HWGPost by Heinz Wrobel

You can find them all on the Aminet.
You have to pay attention that all fonts used by MathScript are
accessable to your interpreter.
If your application does not handle PostScript at all you can still
output your formulas in the common IFF-ILBM bitmap format, the quality
will not be optimal though.

What if I have a PostScript printer?

If you have a printer that has a PostScript interpreter built in, you
have just to pay attention that all fonts you use in your EPS files
are available to your printer.
First of all you should use fonts that are built into the printer like
e.g. Times and Symbol.
But if you also use other fonts like special math fonts, these fonts
have to be downloaded to your printer.
EPS files produced by MathScript have special DSC (Document
Structuring Conventions) comments that tell the importing application
what fonts are nedded. However I am not aware of any program on the
Amiga which takes care of this.
So turn on the "Include Fonts" option in the EPS settings. MathScript
will then include the fonts needed into every EPS file. Because this
will make the files much larger there is a file "FontList" in the
"Files" directory. All fonts listed in this file will not be included.

If you modify this file so that it contains all fonts built into your
printer, only those fonts which really have to be downloaded will be
included.
Another possibility is to deactivate the "Include Fonts" option and to
download the fonts needed manually. You will need a download tool for
this since all binary PostScript fonts (.pfb) have to be decoded
first. This method will probably save you a bunch of memory but you
always have to download the fonts when you have turn on your printer.

Why doesn't FinalWriter or WordWorth display and/or print my formulas
correctly?

Almost all problems like this happen, because the importing
application cannot find the needed fonts used by the formulas. Check
out the following three points:
1. You must have modified init.ps (FinalWriter) or eps_init.ps
   (WordWorth) like explained in Change init.ps.
2. Your fonts have to be in a directory PSFonts: is assigned to.
3. Your font files must have the same name as the font itself (with a
   possible suffix .pfb/.pfa). There are some tools available that can
   rename your fonts automatically.

If this does not help you, you can use a tool like SnoopDos to see why
the fonts could not be opened.

1.56 Updates

Updates

MathScript is in no way a finished project. I will always keep it
improving. Whenever there is a new version ready, I will make it
available at least in the following places:

Aminet:
 The latest version should be available on every Aminet site.
 It will be placed in the misc/math directory, the filename will be
 something like MathScriptXX.lha, where XX is the version number.

MathScript Support BBS:
 "NotreDame"
 Modem: +49-(0)40-247835 (V.34) (6 Ports)
 ISDN: +49-(0)40-280500-32 (X.75)
                        -33
 Login: "Gast"
 Dir: <---Support--->/MathScript

1.57 Credits

Credits

The following people deserve my credits for the development of
MathScript:

  – My beta testers, who helped me a lot with their bug-reports and
    suggestions:

      Chris Coulson
      Kai Nickel
      Guy Tanner

  – Stefan Stuntz for his MagicUserInterface

  – Adrian Aylward for his post.library

  – Kai Nickel for the MathScript support of his programs "Graph2D"
    and "mathX"

  – and last but not least all registered users!

1.58 "

                        This application uses

                    MUI – MagicUserInterface

                (c) Copyright 1993–96 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

Support and online registration is available at

http://www.sasg.com/