

# Release Notes MacMETH 3.2.4

Welcome to MacMETH, a "Fast Modula-2 Language System For the Apple Macintosh".

## Where to Find Help?

Please note, this release contains a folder which holds the entire manual and other useful documentation in electronic form. You have to print this information in order to work successfully with MacMETH. We don't expect you to be able to work with such a neat, yet powerful tool, unless you really consult the manual. We have put much effort into this documentation and hope it will be useful to you.

For feedbacks send an E-mail to [MacMETH@ito.umnw.ethz.ch](mailto:MacMETH@ito.umnw.ethz.ch)  
We like to hear from you. Thanks!

To learn about the newest gossip and get the latest updates visit  
<http://www.ito.umnw.ethz.ch/SysEcol>

And now, be invited to plunge ahead. Enjoy!

## Getting Started Quickly

First acquaintance with the compiler and program execution: Double-click the MacMETH shell, choose the menu command Compile and hit return. The example program module "Example.MOD" will be compiled. Click with the mouse or hit return and you will leave the compiler. Choose the menu command Execute and select the just compiled object file "Example.OBM" (it is in the same folder its source was, i.e. in the folder "Examples"). The

example program will then execute and give you the subsequently needed instructions.

Next compile another program to learn about editing and the handling of compilation errors. In the MacMETH shell choose the menu command Compile again and just type the name of the module "Buggy.MOD". Then hit return. The compiler will compile the program and detect errors. To correct them, choose the menu command Edit2 and you will launch the editor MEdit. Since you do this the very first time, this editor will ask you to provide some customization information. Enter your last and first name where asked and simply accept all other configuration dialogs with OK. The program "Buggy.MOD" will open in a new window and the first compiler error will be displayed in an alert. Correct this syntax error and return to the MacMETH shell by using the menu command "Clear, save and launch" (Cmd^1) in the menu Macros of the editor MEdit. Once in the MacMETH shell compile again and execute as described above.

Should you get stuck or encounter some other difficulties, here we can't help you much more. At the risk to repeat ourselves: Please consult the manual provided in electronic form as a part of this release. This should help. It contains a first tour using the module "Buggy.MOD" in all details plus much more useful information which helps you to make best use of MacMETH.

## The Manual

To obtain the manual simply double-click the file "MacMETH Manual 1992.sea" to unpack it. You will get Microsoft Word documents and a folder containing all sample programs explained in the text. The manual is 116 pages long and you ought to obtain a nice result if you can use a laser or ink-jet printer.

## What's new (not in the manual):

- MacMETH is now supported by 'Systems Ecology' and the Computing Centre ETH Zurich, Switzerland. It is available as freeware courtesy ETHZ via anonymous ftp from the host

ftp.ito.umnw.ethz.ch (Internet node 129.132.80.130)

or on CD as a good bargain from

Celestin Company Inc., 5652 NE Meadow Road, Kingston,

Washington 98346-9505 (Phone: (360) 297-8091/Fax: (360) 297-8092; <http://www.celestin.com/>; For a list of products send and E-mail to [info@celestin.com](mailto:info@celestin.com)

- The memory management has been improved once more considerably:

- The stack size is set to Apple's default stack size (32kB on Color QuickDraw machines and 8KB on non Color QuickDraw machines) plus an additional 10kB.

- However, you can now control the stack size by the User.Profile if you want more than just the additional 10kB. Simply write in the User.Profile the "Stack" section similar to this:

```
"Stack"  
    'Size'           30kB
```

You can enter any number (in kB), provided you give a sufficient total for the application so that there is still some heap memory available. Note, the stack size can only be changed once, i.e. during application start-up. Hence, any stack changes in the User.Profile while the MacMETH, RAMSES shell or Modula-2 application is still running, will not take any effect until you restart this application.

(Note: Size is actually the amount by which the stack is increased in addition to the predefined one. The stack sizes predefined by Apple are: 8 kB for machines without and 32 kB for those with ColorQuickDraw).

- The application heap has been expanded to the maximum possible value. Therefore the Macintosh is now able to detect stack overflows (ID=28). If you frequently encounter such errors, with code which run more or less well under previous MacMETH versions, you have good reasons to suspect run stack overflows. Try increasing the stack size and if this helps, you got the proof. However, now you should check your code, somewhere close to the code area where you get the ID=28 bomb, is the culprit; you should fix it. For instance, check for large objects you pass as actual parameter to an open array value parameter.

- A safety memory cushion (20KB) is allocated at the highest possible position of the heap. This memory will be released if all the other heap memory is used up. The user will be informed, if this happens. This prevents many crashes in the context of ToolBox routines, which are using the application heap.

- The System works now with relative path names. This is a partial work around for the limitation of 63 characters in files names of MacMETH. We were unable to break this limitation, because it appears in the base definition modules and we didn't want to make the entire release key incompatible.

- A new tool, InsertRes, available from within the MacMETH shell has been added. It handles the default access to resources in a more convenient way. This is especially useful if you are using the 'Dialog Machine' or other RAMSES tools. A typical use is the access of 'PICT' resources via procedure 'DisplayPredefinedPicture' from module 'DMWindowIO' which will find the picture resource via the default strategy (i.e. empty file name string). Simply execute InsertRes after the compilation of the (main) program module and the resource will be inserted into the OBM-file; you need no longer to link a double-clickable application to access such resources (for details see document 'InsertRes.DOC' in the folder 'MacMETH Manual 1992').
- Machines with 68LC040 processors are now handled correctly.
- The compilers accept now the old syntax for type transfers in the form T(e), which is the equivalent of SYSTEM.VAL(T,e). Supports portability of source code among different computer-platforms, i.e. Macintosh, IBM-PC, and Unix workstations (for more details see the 'Dialog Machine' and RAMSES, also available via anonymous ftp from the same host given above in the ftp directory 'pub').
- A new method to configure the Modula-2 run-time environment of linked, double-clickable applications. You may omit to use a User.Profile by using a so-called Modula-2 configuration resource instead. The resource of type 'STR' with ID = 7413 defines the following settings (for the meaning see the manual, section 1.4. «The Configuration File "User.Profile"»):

| Entry   | Section      |
|---------|--------------|
|         | Index i      |
| "Traps" |              |
| 'All'   | 1            |
|         | 'Arithmetic' |
|         | 2            |
|         | 'FPU'        |
|         | 3            |
|         | 'F-Line'     |
|         | 4            |
|         | 'System'     |
|         | 5            |
|         | 'Break'      |
|         | 6            |

|        |                         |       |
|--------|-------------------------|-------|
|        | reserved (internal use) | 7     |
|        | "SANE"                  |       |
|        | 'alwaysSANE'            | 8     |
|        | 'invalidHalt'           | 9     |
| 10     | 'underflowHalt'         |       |
|        | 'overflowHalt'          | 11    |
|        | 'divByZeroHalt'         | 12    |
|        | 'inexactHalt'           | 13    |
| 14     | reserved (internal use) |       |
| 15     | reserved (internal use) |       |
|        | "Stack"                 |       |
| 'Size' |                         | 16..n |

A character at index  $i = '1'$  turns the corresponding entry on, any other value

off. If the string is too short to contain a character at position *i*, the corresponding entry is not modified and keeps its predefined default value. The integer number starting at index 16 specifies the stack size in kBytes (followed optionally by any non-digit characters).

Example:

```
110111-110110--10kB
```

indices

```
123456789o123456789..
```

Effect (unless overwritten by a User.Profile residing within the same folder as the Modula-2 application containing the configuration resource): All = on, Arithmetic = on, FPU = off, F-Line = on, System = on, Break = on, alwaysSANE = on, invalidHalt = enabled, underflowHalt = disabled, overflowHalt = enabled, divByZeroHalt = enabled, inexactHalt = disabled; total stack size 18 or 42 kB (depending on machine). Note, these settings are the default settings.

To simulate default settings of older MacMETH's (<=3.2), use the following settings:

Example:

```
110111-010110--10kB
```

indices

```
123456789o123456789..
```

Actually, these are the settings used by the "Linker Extra" (contained in folder M2Tools) which is only the standard linker linked to a stand-alone application, but which inserts above configuration resource. However, note, this emulation of older MacMETH's behavior uses the memory still slightly differently, since there was no top-of heap memory cushion set aside between stack and heap (now 20 kB, to be used if heap gets scarce) and stack overflows were less detected, since the heap was not immediately maximally used (no call of MaxApplZone). Moreover, the stack-heap border line was set at the bottom of the block where now the memory cushion starts.

NOTE: If this resource is missing in the resource file you specify during linking, i.e. your MainModule.R-file, it will be copied into your application from the linking application (typically the MacMETH or RAMSES shell) together with all other essential resources.

## Power Macintosh:

MacMETH doesn't (and will probably never) produce native PowerPC code, but works fine with the 68LC040 emulation of the Power Macintosh machines. So far we found not the tiniest glitch (compliments to Apple)!

If you plan to use the Compile20 version of the compiler, you must have installed the system extension "SoftwareFPU" (actually a control panel or CDEV) by John Neil & Associates, P.O. Box 2156, Cupertino, CA 95015 USA, (Orders: +1(800) 663-2943; Fax: +1 (415) 661-2944; E-Mail: johnneil@netcom.com) and you must also set:

'F-Line'        off

in the Trap section of the User.Profile.

A.Fischlin & J.Thoeny / June.96, ETHZ