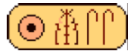


On RAMSES Version 3.0



Research Aids for Modeling and Simulation of Environmental Systems

What is RAMSES?

RAMSES is a huge modeling and simulation environment containing many, many tools useful for a large number of purposes also different from modeling and simulation, such as macro editors, compilers, statistical tools etc. To learn more about RAMSES install it first (instructions below) and read the on-line helps. This RAMSES release does also contain many documentations ready to be printed. RAMSES has a home page at

<http://www.sysecol.ethz.ch/RAMSES/>

Its relationship to our research is described at <http://www.sysecol.ethz.ch/Research.html#RAMSES> and in several publications referenced at <http://www.sysecol.ethz.ch/Publications.html>.

af, dg / ETH Zurich (Last modified 26/07/2006)

You can read this text also after having downloaded the RAMSES software. It is contained in the file "On RAMSES (READ ME)".

How to get RAMSES?

The newest version of the RAMSES software package can be obtained via the Internet at site

<http://www.sysecol.ethz.ch/SimSoftware/>

at no charge (courtesy ETHZ). Alternatively you can also get RAMSES via anonymous internet file transfer ftp (File Transfer Protocol) from the host

<ftp://atitlan.ethz.ch/pub/mac/RAMSES>

For your convenience RAMSES is available in form of compressed files. Our download site at <http://www.sysecol.ethz.ch/SimSoftware/#RAMSES> offers RAMSES as a single file. It is compressed in the standard StuffIt format. Moreover, you don't even need StuffIt Expander to unpack it, since it comes as a so-called self-extracting archive (extension ".sea"). A simple double-click is sufficient to obtain RAMSES in its original form.

How to install RAMSES?

To install RAMSES simply double-click the self-extracting archive you obtained from the download (see also above). Note, the installation means only extracting the original files. Nothing needs to be installed or altered in your system!

Step 1:

To install RAMSES simply double click the file "RAMSES x.y.sea". You will be asked to select a destination folder, into which the folder "RMS" containing the entire RAMSES software package will be written. Select a proper destination and the installation will soon be finished.

Step 2:

For further instructions and informations simply double-click the RAMSES shell. This is the file with the name " RAMSES Shell". There select the menu command "Help...". Have fun!

IMPORTANT NOTICE: RAMSES is only fully functional if executed from a mass storage device onto which it can write. E.g. the compiler needs to write the code it generates or the RAMSES shell attempts to remember the current work object etc. Thus you need first to copy RAMSES always onto an ordinary hard disk and not to double-click it directly on a CD nor on a write protected removable cartridge or other locked disk.

but

Step 3:

Please observe also the following points before actually proceeding with using RAMSES:

- During installation a large number of files (more than 1'000) and folders (more than 20) were generated. They all reside only within your RAMSES folder, i.e. folder "RMS". Note, neither your System folder nor any of your system files have been altered in any way during the installation.
- You may move the folder "RMS" to any volume or location within your file system any time except RAMSES is currently running. However, watch out for the length of the folder names.
- It is recommended to keep the names of the folders in which RAMSES resides very short. If RAMSES is executed deeply nested within folders within folders which have long names, RAMSES is likely to get into troubles. This means it may not be able to access all the files it needs. For upward compatibility reasons with older parts of the RAMSES software, we stick to the shorter path lengths, i.e. 63 characters only, as published by Apple(R) in the mid Eighties.
- The secret for accessing successfully files is to keep the RAMSES folder very close to the root. A pretty good path is:

HD:RMS:

- The next secret for accessing successfully all files is the file with the name "User.Profile". It should reside always within your RAMSES folder. It is used by the RAMSES shell, each time you launch it. Any path specification the shell finds in that file allows it to access files within directories the listed paths point to. The "User.Profile" is a plain text file. You may edit its content to accomodate your specific needs as given by your work. However, don't change its name. Details on how to edit the

"User.Profile" can be found in the MacMETH manual, included in the RAMSES release (see folder "M2" within folder "Docu").

- If you are using MacOS X, the arrangement of items within the RAMSES folder ("RMS") may not look very neat. Then it is recommended to run the AppleScript "Arrange_RMS_X". It comes with the RAMSES Extras release, not the standard RAMSES release You can find RAMSES Extras at http://www.sysecol.ethz.ch/SimSoftware/#RAMSES_Extras. Simply double-click the little utility "Arrange_RMS_X" and you will get a neatly arranged RAMSES folder.
- The folder "RMS" will require about 25 MBytes of disk space. You should have some more space available, if you plan to add large development projects.
- RAMSES will continue to function as long as you do not remove any vital files or folders from its content. In particular be warned not to remove or rename any folders containing binaries (files with extensions .SBM or .OBM) from the folder "RMS" unless you know exactly what you do. To learn all details about RAMSES' finer points, you need to study the manuals contained in this package carefully (see below).
- If you start the editor " MEdit 1.80" the first time, e.g. via selecting the menu command "Edit" from within the RAMSES shell, please follow the configuration dialog by entering your last and first name and clicking 'Ok' for all the rest of the dialogs. This allows you during all future uses of the RAMSES shell to easily switch back and forth between the shell and MEdit via the shell's menu command "Edit..." respectively the command "Clear, save & launch" in the editor's menu "Macros".
- RAMSES should run on any Apple(R) Macintosh(R) computer run under System software later than Finder 5.3 and System 3.2 (including the newest system software such as MacOS X in the Classic environment). The memory requirements depend on what you do. For large modeling and simulation projects, you may require many Mega Bytes of free RAM, small RAMSES applications may require less than half MByte RAM. It is recommended to give the shell " RAMSES Shell" at least 1.5 MBytes working memory, and the default editor " MEdit 1.80" 640K Bytes (use Finder menu command Get Info, tab memory).
- The available on-line help contains also references to further documentation: First to the documentation distributed in electronic form as part of the RAMSES software package and second to published material, e.g. scientific articles (<http://www.sysecol.ethz.ch/Publications.html>), describing RAMSES further (see also section Documentation). Technical details on the installation and implementation of RAMSES are contained in "RAMSES Install&Tech Report.pdf"



Documentation

RAMSES is released with plenty of documentations in various formats.

Documentations are mostly in the folders "Docu" and "Help Files".

There is also an internet based reference available at

<http://www.sysecol.ethz.ch/RAMSES/Objects/>

Documentations in the folder "Docu" may be consulted while editing by using text editors such as MEdit (included) or Alpha. Some of the documentation files are best opened with SimpleText or NotePad, and all the pdf-files require Adobe Reader. You can download Adobe Reader for free at <http://www.adobe.com/products/acrobat/readermain.html>. Note, the editor Alpha with its mode M2 is particularly useful. It supports to access specific documentation on individual modules and procedures. Read the on-line Help of the M2 mode from within Alpha to learn about the technique to accomplish this (see also section [Latest Changes](#) in this README file).

Documentations in the folder "Help Files" provides the on-line help accessible during program execution. While you use a particular RAMSES session or the shell, the menu command "Help" (or key if available) will open in a context sensitive manner a window. The latter contains the on-line help together with many useful cross references to the documentations contained in folder "Docu", publications available on the internet, and other hints. All this documentation should help you to make good use of the RAMSES software.

Note, the README files and the so-called quick reference files like "DM Quick reference", "MW Quick references", "AuxLib Quick reference", "ISIS Quick Reference", "AuxLibExtra Quick Reference", and "ScienceLib Quick Reference" as well as the files holding the definition modules (DEF-files) are the decisive authority in case of doubts. They describe the latest changes. The README files contain normally a section "Latest changes" describing what can not be found in the aforementioned documentation files (see e.g. section [Latest Changes](#) in this README file).

In case you encounter difficulties, a good starting point for trouble shooting is the section in the corresponding section in the README files (see e.g. section [Known Problems](#) in this README file).

It is highly recommended to print the large manuals (about 500 pages) included in this package. They were designed for this. This is the case for the MacMETH manual (in folder M2 within folder Docu), the "Dialog Machine" documentation (in folder DM within folder Docu), and for ModelWorks (in folder MW within folder Docu).

Most documentation is also available on the Internet. Each software layer of RAMSES has its own home page, where this information can be accessed. The recommended starting point is the home page of RAMSES you find at <http://www.sysecol.ethz.ch/RAMSES/>.

License - What you are entitled to do with RAMSES?

RAMSES is freeware but NOT public domain. This means two things: First, all rights are reserved and copyrights remain with the authors and Swiss Federal Institute of Technology Zürich, Switzerland. Second, the RAMSES software may not be altered or modified by any means.

You may copy RAMSES as many times you wish. However, you may redistribute it only in a completely unaltered form and may not distribute it for profit or as an incentive to buy a product. You may copy this distribution freely

BUT NOT FOR PROFIT!

If you develop a piece of software by means of our base software such as MacMETH, the Dialog Machine, ModelWorks, ISIS, or RAMSES, we require that you give us credit by mentioning in your publications, software and/or documentation in a well visible fashion that you have developed your software by means of the RAMSES package. In particular do we ask you to leave the copyright notice window of the Dialog Machine in your software intact.

Disclaimer

The authors of the computer software RAMSES hereby disclaim any and all guarantees and warranties on the software or its documentation, both expressed or implied. No liability of any form shall be assumed by the authors. Any user of this software uses it at his or her own risk.

This product is distributed on an "as is" basis; no fitness for any purpose whatsoever nor warranty of merchantability are claimed or implied.

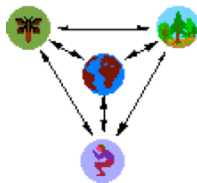
The authors reserve the right to make changes, additions, and improvements to the software or documentation at any time without notice to any person or organization; no guarantee is made that further versions of either will be compatible with any other version.



Feedback

If you wish to contact the authors of RAMSES please do so by using electronic mail only. Send messages to the following internet address: [mailto:RAMSES\[AT\]env.ethz.ch](mailto:RAMSES[AT]env.ethz.ch)

Thank you!



RAMSES Extras

There is also some extra software available beside the ordinary RAMSES package (i.e. all the software contained in the folder "RAMSES x.y"). This add-on software is called **RAMSES x.y Extras**, released as a separate package (You find it's home at http://www.sysecol.ethz.ch/RAMSES/RAMSES_Extras.html).

RAMSES x.y Extras contains software useful if you wish to extend RAMSES further. It offers also many convenient or powerful utilities and tools including shareware from third parties. It is particularly useful for advanced users of RAMSES. For instance, if you don't like MEdit (the free editor, which comes with RAMSES), here you find an alternative support for the shareware editor Alpha (<http://www.kelehers.org/alpha/>). Moreover, you find several additional modules for special purposes, such as low level access to basic mechanisms, e.g. debugging of garbage collection and heap management problems etc. See the definition modules for details.

All this software contained in **RAMSES x.y Extras** is available on a purely as-is basis. Note, this software may be partly in an experimental stage. The freedom to decide whether it will be further supported or even abandoned remains completely with the authors. Nevertheless, one or the other piece may be useful to you.

Enjoy!

Latest Changes

• RAMSES Objects Reference

As of RAMSES 3.4.0r1 an internet based reference documentation on all objects provided by RAMSES is now available at

<http://www.sysecol.ethz.ch/RAMSES/Objects/>

This reference is also available as a download to store locally on your system for off-line viewing. Note, the M2 mode of AlphaX (<http://alpatcl.sourceforge.net/wiki/>) fully supports this reference. For instance while editing some Modula-2 source code from within AlphaX do

Commad-double-click

onto e.g. a module or procedure identifier. The M2 mode should bring you to a quick reference of that object.

Alternatively, if you wish to go to the full DEFINITION of an object, do a

Control-Commad-double-click

onto the identifier of a RAMSES object. The DEFINITION module exporting that object will be opened and your view will be brought directly near the definition of the object. Typically you find there comment text explaining the behavior and use of the object. A most useful feature for any RAMSES programmer!



• Support for AlphaX

[AlphaX](#) runs native in OSX, whereas RAMSES still needs to be run within Classic. Yet, we have managed to support the full collaboration between all these software components. However, you need the latest versions of several software components, in order to smoothly switch back and forth between RAMSES and AlphaX. Use the following table to ensure you have all needed software components installed:

Software Component	Minimum Version	Link
OS X	10.2.x	http://www.apple.com/macosx/
Classic (within OS X)	9.2.2	http://www.apple.com/support/macos9/ http://docs.info.apple.com/article.html?artnum=75288 ftp://atitlan.ethz.ch/pub/mac/RAMSES/
AlphaX	8.0b8	http://www.maths.mq.edu.au/~steffen/Alpha/AlphaX/ http://alphaltcl.sourceforge.net/wiki/pmwiki.php/Software/AlphaX http://alphaltcl.sourceforge.net/wiki/
Tcl/Tk	8.4.2.0	http://www.maths.mq.edu.au/~steffen/tcltk/TclTkAqua/ (no longer needed separately with recent versions of AlphaX, where this component is integrated into the application bundle.)
tclAE	2.0b14	http://www.maths.mq.edu.au/~steffen/weblog/archives/updates_to_tclkaquabi_8420.php contained in the latest M2 Mode package (see below) (no longer needed separately with recent versions of AlphaX, where this component is integrated into the application bundle.)
M2 Mode	3.9.0	ftp://atitlan.ethz.ch/pub/mac/RAMSES/ (no longer needed separately with recent versions of AlphaX, where this component is integrated into the tcl folder accompanying the application.)
RAMSES	3.0.7	http://www.sysecol.ethz.ch/SimSoftware/

Of course you can continue to use RAMSES and Alpha within a native MacOS 9.2.2 system. For such a system we recommend you use the same software components as listed in above table, except for Alpha. AlphaX is of course not available and we recommend to use Alpha 7.6.1 or Alpha8 instead. Please note, Alpha8 has been in use for a relatively short period and thus you may favor Alpha 7.6.1 for its reliability. However, Alpha8 offers a markedly increased speed and greater comfort.



• Improved Numerics and Portability Between RAMSES and RASS

The numerics have been improved once more by providing default floating point arithmetics, which offer highest precision (with least significant bit precision) as well as highest portability at the same time. RAMSES and RASS as of versions 3.0.6 generate for almost all applications and circumstances identical results except for a few exotic cases.

We have thoroughly tested and compared for a large test suite the behavior of many numerical algorithms and found identity in the results for all models, even as complex ones as ForClim. Only one test program, designed to test very critical behavior manages to produce differences among the Macintosh and Sun platforms. Yet alle the remaining absolute or relative deviations stay in within the range of the rounding precision of the involved commputing engines.



• ISIS - Integrative Systems Implementation Software



ISIS has been added to RAMSES.

ISIS supports the modeling of structured systems in a much more efficient manner than this is possible with ModelWorks. Notably data and structure may be separated fully. Data such as initial values or model parameters may be assigned at run time from data frames. Current data may anytime be saved to disk and stored in a data frame file for documentation or later use in a simulation experiment.

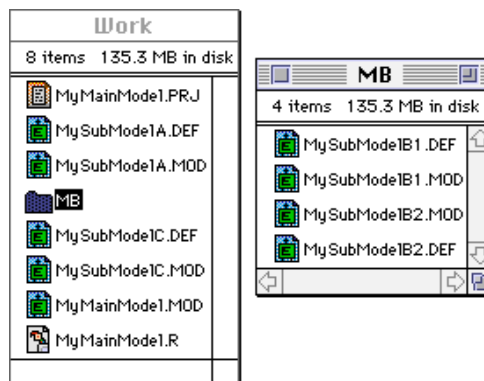


• Better Support for Structured Models via Project Files

Projects may now also refer to files distributed over several folders. The work object's root path is the folder in which the ".PRJ" files resides. Any module listed in this file may be preceded by a relative path. The relative path is understood relative to the work object's root path. Thereafter the "Mini RAMSES Shell" or the "Modeling Session" from the big "RAMSES Shell" will recognize those files and compile them if needed during any making of the work object (either menu command "Simulate" in the "Mini RAMSES Shell" or "Make" in "Modeling Session" of the big shell). E.g. your project file "MyMainModel.PRJ" may look like this:

```
MyMainModel.PRJ
MySubModelA Def Mod;
:MB:MySubModelB1 Def Mod;
:MB:MySubModelB2 Def Mod;
MySubModelC Def Mod;
MyMainModel Mod Res;
```

where your file structure looks similar to this



In above example the work object's root path is the path leading to the folder "Work". In the project file the path of the files "MySubModelBx" is given relative to folder "Work", i.e. ":MB:".

As soon as you change any of the definitions or the project file itself, all modules will be recompiled. If you change only particular implementations, only those affected by the changes will be recompiled. Resources will be automatically added into file MyMainModel.OBM and are ready for access during program execution (method file name empty).



Known Problems and Solutions

There are no known bugs in the latest RAMSES software. However, you may occasionally encounter difficulties, in particular should you be using an older software version and/or be working on older computer systems. The following items are intended to help you to overcome such difficulties.

The following items describe known problems or bugs and hints on how to work around them in RAMSES Versions $\geq 2.2.3$:



• You wish to call the debugger at run time

While your program is running simply press the keyboard combination **Cmd ^ Shift ^ H** and you will be offered a dialog, a so called "**User Halt**", which lets you activate the debugger. Thus there is no need to program always the debugger activation by inserting into your code the statement HALT. However, the latter is of course still the best method, if you wish to have maximum control over where the debugger is activated.



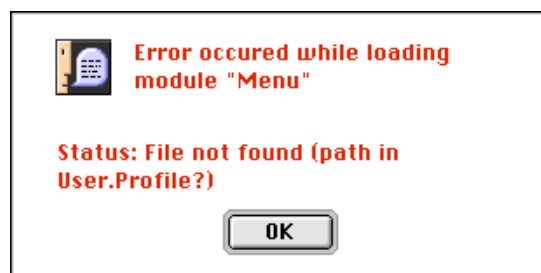
• Event of unknown class encountered after having removed and redeclared the same DEVS model during simulation.

Remedy: This is an implementation restriction for DEVS, since DEVS require globally unique state events at all times in state Simulating. If you remove and redeclare a model during the same time step, the current implementation will detect a conflict, since the old model and its associated event classes are still known to ModelWorks (despite your call to procedure RemoveM, since its execution is actually postponed till the entire integration step is completed for all models, see ModelWorks Manual Table T4, p.T74, section 'Manipulating the model base at run-time^a'). As a consequence the redeclaration will fail (as simulationist you are going to be informed about this fact), but not the removal, which will take place as soon as possible. Thus during all subsequent integration steps, the old as well as the new model will not be known to ModelWorks and events generated for that model will have no "listening" state transition functions. A possible work around is to defer the redeclaration of the DEVS to the next time step, after the removal has actually taken place, or to use completely different event classes for the redeclared model, albeit the associated state transition functions may be the same. Please, let us know if that should not be sufficient (<mailto:RAMSES@env.ethz.ch>) and we will try to find a better solution.



• Files can't be accessed (various error messages, in particular File not found (path in User.Profile?) or Path too long?)

If you get an error message similar to this



you have probably encountered one of the problems described below.

In general the User.Profile's PATH section offers a versatile, simple mechanism to access files (see MacMETH for a detailed description on how to use this mechanism). However, any glitch in the textual definition of paths and the actual file system may easily result in an often confusing erroneous behavior of your software with error messages as the one shown above. In particular:

Remedy:

First, always check that

- **path specification and actual folder names match exactly each other!**

In particular watch out for blanks, hard blanks or other special characters in the names of folders (or files). Such characters may sometimes be invisible, yet they have of course a profound effect on the functioning of file accesses. In rare cases only the careful scanning through folder and file names e.g. with the cursor keys, can reveal such "hidden" characters. Any mismatch between path specification and actual folder names, e.g. due to a tabulator, leads then to a non-functioning file access.

Secondly,

- **make sure you run your RAMSES shell or your linked application from a folder which has a short path!**

RAMSES still runs on older Macintosh machines, i.e. Macintosh Reflex (= Macintosh 512K) or a 1 MB RAM Macintosh Plus using System 3.2. To maintain this upward compatibility we have never supported longer path names than those supported by the limit, which was originally announced by Apple. This limit amounts to 63 characters for a path plus 31 characters for the additional file name. Many algorithms require you to use internally often absolute paths to access files. E.g. a downloaded and unstuffed RAMSES folder with a base path starting with "Macintosh HD:Desktop Folder:RAMSES Folder 2.2.3b48:RAMSES 2.2.3b48:" already exceeds this limit (path alone is 67 characters long) and working with RAMSES in such a folder is not possible. However, you can easily prevent such problems, if you shorten the name of the RAMSES folder, e.g. to "RMS", omit the use of the embedding folder "RAMSES Folder 2.23b48" entirely, and move the folder "RMS" away from the desktop close to the root. It is also recommended to rename the hard disk to the short name "HD". In above example you would get a base path like

HD:RMS:

which is now relatively short and unlikely to cause much problems by exceeding the 63 character limitation; pursuing projects in complex folder structures, e.g. within the folder "HD:RAMSES:Work:", is now possible.



- **Failing DMFiles.Lookup called by an application which resides on the desktop**

Lookup does not properly function when the calling application, e.g. your Modula-2 program linked into an application, sits on the desktop. Files can't be properly accessed in this case, regardless of path specifications in the User.Profile (returned Response is always filenotfound or notdone). This is a known bug.

Partial Remedy: Please move your application to a different folder, where the exactly same application should now work.



- **Can't launch Alpha from the RAMSES shell.**

Sometimes the RAMSES shell and Alpha seem just not be able to communicate with each other properly without apparent reason, because you installed all according to the rules.

Remedy: Trash the files `err.ALPHA` and `token.ALPHA`, recompile and retry. That usually helps. Otherwise check your configuration, in particular the name and folder paths given in the `User.Profile`. Note, they must exactly match your installation (watch for hard-blanks vs. ordinary blanks in path and file names etc.).

If you are using AlphaX make also sure that you add the invisible extension `.app`. E.g. in the section "Alias" of your `User.Profile` you should have an entry similar to this

```
'Alpha' is 'HDX:Applications:Alpha:AlphaX.app'
```



• **Compiler error "output file not opened (directory full?)"**

while the "Modeling" session is trying to make your work object ready for simulation. Usually there appears actually nothing to be wrong, i.e. no errors in your source code and plenty of disk space still available, and your directory is probably not full.

Remedy: In fact, a more likely cause is that the compiler can't overwrite the old OBM-file, since its resource fork is still kept open by some program you have been running under RAMSES, happens especially if you crashed. A simple remedy is to drag the OBM-file to the trash (albeit you may not be able to really trash it unless you restart your machine) or quit the RAMSES shell and relaunch it. That usually fixes this problem. Otherwise if all this still does not help, your directory might actually be full, too many files. Try moving some files to another disk (subsequently don't forget to delete them at the old location to actually free some directory entries).



• **System freeze on Power PCs if you attempt to halt a program via the interrupt (Cmd ^ Power key).**

You can't use the above keyboard short-cut if your hard disk has NOT been formatted by Apple's "Apple HD SC Setup", regardless of Apple's claim that this is part of the System software (see Owner's Manual). This interrupt handler actually sits in the hard disk driver and can therefore be different from system to system, though you're using exactly the same system software.

Remedy: Use a keyboard shortcut to trigger a so-called User Halt. This is a keyboard handler in your program which calls a halt while e.g. simultaneously pressing `Cmd ^ Option ^ H`. Note, such a keyboard handler is available in all MacMETH and the Dialog Machine releases of the RAMSES software with version $\geq 2.2.3$, given you use the U.S. keyboard layout. Later versions, i.e. RAMSES $\geq 3.0.2$, feature also a better shortcut, which is independent from the keyboard layout. It is: **Cmd ^ Shift ^ H**.



• **Under MacOS 8.5 to 9.1 or X certain run time errors, e.g. index range errors, crash the OS.**

Ordinary integer arithmetic exceptions, i.e. division by zero, range checks, and overflow crash your MacOS.

Remedy: Upgrade to the latest RAMSES ($\geq 3.0.2fc5$ / MacMETH 3.2.8) and recompile all your code; or stick to OS 8.1, or install at least MacsBug and disable all traps in the "Traps" section of the "User.Profile". The latter allows you to safely exit to the shell (type command "es" and return) and relaunch MacMETH, avoiding the cumbersome restart of the OS. MacsBug is available as a free software download from <http://developer.apple.com/tools/debuggers/MacsBug/>

or <http://www.Apple.com>. See also README MacMETH you find at http://www.sysecol.ethz.ch/SimSoftware/RAMSES/READ_ME.MacMETH.html for further details on this issue and the latest changes relative to the MacMETH manual on exceptions, traps, and run-time error control.



- **Under MacOS 8.5 to 9.0.4 the break-point debugger crashes the OS.**

This is probably a problem we won't be able to fix, since Apple seems not to officially support proper installation of interrupt handlers dealing with traps, which are usually handled by Apple as so-called system errors, resulting in the highly appreciated bombs ;-). In case anyone should know about how and where to install interrupt handlers, please let us know and we will reintroduce this nice feature, i.e. the setting of break-points, under any MacOS. However, as long as Apple keeps the needed information as a secret, we won't be able to offer anymore a proper solution.

Remedy: Upgrade to MacOS 9.1 or X, or stick to OS 8.1, or install at least MacsBug. MacsBug is available as a free software download from

<http://developer.apple.com/tools/debuggers/MacsBug/>

or <http://www.Apple.com>. See also README MacMETH you find at http://www.sysecol.ethz.ch/SimSoftware/RAMSES/READ_ME.MacMETH.html for further details on this issue and the latest changes relative to the MacMETH manual on exceptions, traps, and run-time error control.



- **You are using MacOS 9 and get a system crash when you try to launch an editor such as MEdit or Alpha, e.g. by choosing the menu command "Edit..." as offered by the RAMSES Mini shell and many sessions in the RAMSES big shell.**

Remedy: Upgrade to the latest version of RAMSES, in particular a version greater than RAMSES 3.0. There the bug (by Apple) causes no longer a system crash and you can again smoothly switch between the RAMSES shell and the editor.



- **Help (see module in AuxLib of this name) does not recognize your help file and insists on giving you only an error message while attempting to read it.**

Remedy: Make sure you generated the file using Microsoft Word 5.1 or even older versions. RTF-files produced with Microsoft Word 6 and newer are not properly recognized by the current version of the RTF scanner used by the module Help. Since you can generally save files in the older formats, e.g. with Word 6' "Save as" command, it should be easy to work around this problem. Actually, I find editing help files using Word 5.1 more convenient, since it still shows you the borders of the page with the size of the window you actually have to fit your text and graphics in. Word 6 doesn't do that anymore.



- **System bombs (illegal instruction) on Power PCs while launching the compiler "Compile20" or the execution of code generated by this compiler.**

Remedy: PowerPCs 68K emulator does only emulate a 68LC040 not a true 68040. Hence it has no FPU built into the emulator. In order to execute such code you need to enhance the 68K-emulation, e.g. with the extension "SoftwareFPU" or "PowerFPU" (details on availability and settings are given in "README MacMETH" in the section Power Macintosh).

Note, in general it is **not** recommended to use Compile20 on current generation of Macintosh computers. It offers no longer any speed gains. Instead you should stick to the compiler **Compile** and make sure it is at least version 2.6.8.



- **If you notice a degradation in the performance of Compile20 generated code on a Power Macintosh**

Remedy: You can improve the floating-point performance by one or all of the following measures:

- 1. Use "PowerFPU" instead of "SoftwareFPU" (see hint above).
 2. Use SANE implementations wherever available (e.g. rename
 3. "DMMathLib.OBM (SANE)" to "DMMathLib.OBM", in future releases of RAMSES you will find also the "DMMathLF.OBM (SANE)" implementation, which could also be renamed to "DMMathLF.OBM"). 3) Compile your code with the ordinary Compile compiler and set the alwaysSANE flag in the "SANE"-section of your "User.Profile" to off.
 4. In general, (i) basic floating point operations (+ - * /) are slightly faster when executed as Compile20-code instead of using SANE and fastest executed as Compile-code with alwaysSANE turned off and (ii) transcendental functions are fastest executed using SANE as Compile-code (however, this may depend on your actual system configuration, but was found to be true on PPC 604e machines, running under System 7.5.5 and when using PowerFPU 1.2)
 5. Turn off all extensions (except those needed for the mouse, the monitor, and "SoftwareFPU" or "PowerFPU" respectively) and you will get a lot of performance improvement. Apple's implementation of SANE tends to get more efficient, but the System software tends to get at the same time slower, so that only turning extensions off lets you access the gain in speed of SANE (may vary depending on your actual system configuration, but was found to be true when comparing System 7.5.3 Rev.2 with System 7.5.5).

However note, it is **not** recommended to use Compile20 on current generation of Macintosh computers. It offers no longer any speed gains. Instead you should stick to the compiler **Compile** and make sure it is at least version 2.6.8.

Last modified 26/07/2006 by Andreas Fischlin, ETHZ

[RAMSES Home](#) | [RAMSES Download](#) | [Systems Ecology](#) | [ETH Zurich](#)