

# **Executive**

Petri Nordlund

Copyright © CopyrightÂ©1995-96 Petri Nordlund. All rights reserved.

---

**COLLABORATORS**

	<i>TITLE :</i> Executive		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Petri Nordlund	April 16, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Executive</b>	<b>1</b>
1.1	Main menu	1
1.2	...	2
1.3	Copyright and Disclaimer	2
1.4	Acknowledgments	5
1.5	History	7
1.6	Introduction	18
1.7	Getting started	19
1.8	Multitasking basics	20
1.9	Different kinds of tasks	21
1.10	Scheduling	22
1.11	Executive features and requirements	24
1.12	How does it work?	26
1.13	Patched functions	28
1.14	Installed files	28
1.15	Moving Executive to another directory	29
1.16	Common problems	30
1.17	Tutorial	34
1.18	Server	38
1.19	Starting the server	38
1.20	Making it work	38
1.21	Server options	41
1.22	Commodity	44
1.23	Quitting the server	44
1.24	Preferences file	45
1.25	RELATIVE	47
1.26	Server Internals	48
1.27	Clients	52
1.28	ScreenNotify	55
1.29	Genies	56

---

---

1.30 Schedulers . . . . .	57
1.31 Accounting . . . . .	59
1.32 Registration . . . . .	61
1.33 Keyfile . . . . .	64
1.34 Glossary . . . . .	64
1.35 Developers . . . . .	66
1.36 Author . . . . .	66
1.37 PGP KEY . . . . .	67
1.38 Background . . . . .	68
1.39 Acct . . . . .	68
1.40 Acct options . . . . .	70
1.41 ALoad . . . . .	72
1.42 ALoad options . . . . .	73
1.43 ALoad3D . . . . .	77
1.44 ALoad3D options . . . . .	78
1.45 Commander . . . . .	81
1.46 Commander options . . . . .	83
1.47 Ctp . . . . .	84
1.48 Ctp options . . . . .	85
1.49 Dashboard . . . . .	86
1.50 Dashboard - Introduction . . . . .	87
1.51 Dashboard - Getting started . . . . .	88
1.52 Dashboard - Common elements . . . . .	89
1.53 Dashboard - Global settings . . . . .	92
1.54 Dashboard - Windows . . . . .	93
1.55 Dashboard - Meters . . . . .	94
1.56 Dashboard - CPU usage . . . . .	98
1.57 Dashboard - Recent CPU usage . . . . .	98
1.58 Dashboard - Load average . . . . .	99
1.59 Dashboard - Total used CPU time . . . . .	99
1.60 Dashboard - Total idle CPU time . . . . .	100
1.61 Dashboard - Context switches . . . . .	100
1.62 Dashboard - Tasks running/sleeping . . . . .	101
1.63 Dashboard - CPU usage . . . . .	101
1.64 Dashboard - Recent CPU usage . . . . .	102
1.65 Dashboard - Used CPU time . . . . .	102
1.66 Dashboard - Used CPU time (+children) . . . . .	102
1.67 Dashboard - Child CPU . . . . .	103
1.68 Dashboard - Context switches . . . . .	103

---

---

1.69 Dashboard - Stack . . . . .	104
1.70 Dashboard - Memory . . . . .	105
1.71 Dashboard - Disk usage . . . . .	105
1.72 Dashboard - Device I/O . . . . .	106
1.73 Dashboard - Drive light . . . . .	107
1.74 Dashboard - VMM stats . . . . .	107
1.75 Dashboard - Clock . . . . .	108
1.76 Dashboard - Date . . . . .	108
1.77 Dashboard - Calendar . . . . .	108
1.78 Dashboard - On-line . . . . .	109
1.79 Dashboard - SANA-II status . . . . .	109
1.80 Dashboard - SANA-II status (LED) . . . . .	110
1.81 Dashboard - Textbox . . . . .	111
1.82 Dashboard - Uptime . . . . .	111
1.83 Dashboard - Users . . . . .	111
1.84 Dashboard - Round . . . . .	112
1.85 Dashboard - Analog . . . . .	112
1.86 Dashboard - Graph . . . . .	112
1.87 Dashboard - Auto graph . . . . .	113
1.88 Dashboard - Bar . . . . .	113
1.89 Dashboard - Fuel . . . . .	114
1.90 Dashboard - LED . . . . .	114
1.91 Dashboard - Text . . . . .	114
1.92 Dashboard - Text . . . . .	114
1.93 Dashboard - Text . . . . .	115
1.94 Dashboard - Text . . . . .	115
1.95 Dashboard - Text . . . . .	115
1.96 Dashboard - Scale . . . . .	116
1.97 Dashboard - ScreenNotify . . . . .	117
1.98 Dashboard - Tips & tricks . . . . .	118
1.99 Dashboard options . . . . .	119
1.100ExecutivePrefs . . . . .	120
1.101ExecutivePrefs options . . . . .	123
1.102Kill . . . . .	125
1.103Kill options . . . . .	126
1.104Lastcomm . . . . .	128
1.105Lastcomm options . . . . .	129
1.106Meter . . . . .	130
1.107Meter options . . . . .	133

---

---

1.108Nice . . . . .	138
1.109Nice options . . . . .	139
1.110Ps . . . . .	140
1.111Ps options . . . . .	142
1.112Ps Keywords . . . . .	148
1.113Pstree . . . . .	154
1.114Pstree options . . . . .	156
1.115Renice . . . . .	157
1.116Renice options . . . . .	158
1.117Sa . . . . .	160
1.118Sa options . . . . .	162
1.119Timer . . . . .	165
1.120Timer options . . . . .	167
1.121Top . . . . .	169
1.122Top options . . . . .	172
1.123Stat . . . . .	176
1.124Stat options . . . . .	178
1.125Uptime . . . . .	178
1.126Uptime options . . . . .	179
1.127About custom shells . . . . .	182

---

# Chapter 1

## Executive

### 1.1 Main menu

E x e c u t i v e

Copyright

Acknowledgments

History

Introduction

Common problems

Tutorial

Server

Clients

Schedulers

Accounting

Registration

Glossary

Developers

Author

Background

---



Acct  
Dashboard  
Nice  
Stat  
ALoad  
ExecutivePrefs  
Ps  
Timer  
ALoad3D  
Kill  
Pstree  
Top  
Commander  
Lastcomm  
Renice  
Uptime  
Ctp  
Meter  
Sa

by Petri Nordlund

## 1.2 ...

This space was intentionally left blank.

## 1.3 Copyright and Disclaimer

COPYRIGHT

~~~~~

Copyright © 1995-96 Petri Nordlund.

Executive software and logo are Copyright © 1995-96 Petri Nordlund.  
All rights reserved.

#### LICENCE

Executive is not, nor has ever been, public domain or free software.  
Users are encouraged to register, not enforced.

You are granted the right to share Executive with others, as long as  
you distribute the Executive archive exactly as you received it, with  
all associated files included. REGISTERED USERS MAY NOT DISTRIBUTE  
THE SEPARATE FILE "Executive.key".

Under no circumstances may you charge more than a normal copying fee  
and shipping costs for distributing the Executive files without  
express written consent from the copyright holder.

Bulletin board system operators may post the unregistered Executive  
software on their BBS for downloading by their users without written  
permission only if the above conditions are met, and only if no special  
fee is necessary to access the Executive files. A general fee to access  
the BBS is ok.

Reverse engineering of the Executive software protection is strictly  
forbidden.

Several digital fingerprints are included in the "Executive.key" file  
and redistributed copies are easy to identify. Any redistribution of  
this file will lead to legal actions.

Distributing beta-versions of Executive is strictly prohibited.

Executive logo can be used for advertising purposes.

#### DISCLAIMER

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND  
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR  
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS  
BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR  
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF  
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS  
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN  
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)  
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF  
THE POSSIBILITY OF SUCH DAMAGE.

#### GADLAYOUT

Commander, ExecutivePrefs and Timer make use of the GadLayout dynamic

gadget layout system by Timothy Aston.

MUI

~~~~~

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/>

MAGICWB

~~~~~

The MagicWB is Copyright © 1992-94 Martin Huttenloher

TRADEMARKS

~~~~~

Amiga and Workbench are trademarks of Amiga Technologies GmbH.

VAX/VMS is a trademark of Digital Equipment Corporation.

UNIX is a trademark of X/Open Company Ltd.

OS/2 is a trademark of International Business Machines Corporation.

Windows 95 and Windows NT are trademarks of Microsoft Corporation.

SAS/C is a trademark of SAS Institute Inc.

Postscript is a trademark of Adobe Systems Inc.

Final Writer is a trademark of SoftWood Inc.

Macintosh is a trademark of Apple Computer, Inc.

## 1.4 Acknowledgments

### ACKNOWLEDGMENTS

~~~~~

Thanks to all betatesters and catalog translators:

#### Betatesters

|                   |                                        |
|-------------------|----------------------------------------|
| Lauri Aalto       | <aalto@iki.fi>                         |
| Osma Ahvenlampi   | <oa@iki.fi>                            |
| Doug Dyer         | <dyer@alx.sticomnet.com>               |
| Pascal Eeftinck   | <arcade@xs4all.nl>                     |
| Georges Goncalves | <morgoth@club-internet.fr>             |
| Mike Hellers      | <hellers_m@istvax.ist.lu>              |
| Andreas Jung      | <ajung@hssun5.cs.uni-sb.de>            |
| Mathias Karlsson  | <d3karma@ttek.chalmers.se>             |
| Wolfgang Loske    | <wloske@cip.e-technik.uni-erlangen.de> |
| Koen Peetermans   | <Koen.Peetermans@ping.be>              |
| Marcin Orlowski   | <carlos@dedal.man.szczecin.pl>         |
| Antonio Santos    | <L38058@alfa.ist.utl.pt>               |
| Christoph Stoppe  | <cstoppe@piezke.prima.ruhr.de>         |
| Teemu Suikki      | <zuikkis@zuikkis.pp.sci.fi>            |

#### Catalog translators

|            |                      |                                  |
|------------|----------------------|----------------------------------|
| Croatian   | Jovica Popovic       | <Jovica.Popovic@vidikon.fido.hr> |
| Dutch      | Pascal Eeftinck      | <arcade@xs4all.nl>               |
| Finnish    | Petri Nordlund       | <petrin@megabaud.fi>             |
| French     | Julien Wilk          | <wilkj@esiee.fr>                 |
|            | Georges Goncalves    | <morgoth@club-internet.fr>       |
| German     | Christoph Stoppe     | <cstoppe@piezke.prima.ruhr.de>   |
| Greek      | Stelios D.Melissakis | <djmelis@hol.gr>                 |
| Italian    | Stefano P.R. Peruzzi | <peru@maya.dei.unipd.it>         |
| Norwegian  | Torbjørn Burchard    | <torbb@icenet.no>, Kjell Irgens  |
| Polish     | Marcin Orlowski      | <carlos@dedal.man.szczecin.pl>   |
| Portuguese | Antonio Santos       | <L38058@alfa.ist.utl.pt>         |
| Swedish    | Staffan Lindberg     | <pike@mailbox.swipnet.se>        |
|            | Mathias Karlsson     | <d3karma@ttek.chalmers.se>       |

I also want to thank the following persons:

Osma Ahvenlampi

for ideas (SysInfo.library).

All Aminet moderators and maintainers  
for Aminet!

Timothy J. Aston  
for GadLayout.

Mat Bettinson  
from the #1 Amiga magazine.

Pascal Eeftinck  
for many ideas and comments.

Nico François  
whose Cat2H utility solved all the problems I had with catalogs.

HiSoft  
I have used their Devpac assembler for many years now and have  
been fully satisfied. Competiton still comes far behind.

Thomas Igracki  
for SysInfo.library Amiga-Oberon interface.

Wolfgang Loske  
for numerous ideas and comments.

Goran Paulin  
for accepting registrations in Croatia.

Richard Stallman & Roland McGrath  
and others for GNU Make.

Stefan Stuntz  
for MagicUserInterface.

Teemu Suikki  
for ideas and comments.

Geert Uytterhoeven  
for MultiUser.

Philip A. Vedovatti  
for drawing Executive NewIcons.

Markus Wild, Philippe Brand, Lars Hecking, Fred Fish & Leonard Norrgard  
and others for porting GCC to AmigaOS.

Thanks to everybody who has supported Executive development by  
registering Executive.

I have received over 6 MB of Executive related email. I want to thank  
everybody for suggestions, ideas and bugfixes, keep them coming!

Without all of you, Executive wouldn't have been possible.

---

## 1.5 History

### HISTORY

```

~~~~~
V1.00
V1.10
V1.20
V1.30
V2.00
V1.00
~~~~~
First public release.

```

#### V1.10

```

~~~~~
* Top-client now supports VMM. It displays available virtual memory if
  VMM has been installed.

* Added ToolsDaemon And VLT/VLTjr to the Magic Wand list.

* The server now defaults to using CIA-B timer on 68020+ machines and
  CIA-A timer on 68000 and 68010.

* Added a Common problems section to docs.

* A .Product-Info file is now included.

* ExecutivePrefs[_MUI] will now check for duplicate entries in task-list.

* If a new entry is created in ExecutivePrefs[_MUI] by pressing the
  New-gadget but then Cancel-gadget is pressed in Edit-window, the
  entry will be deleted.

* When the initial preferences file is written, system task list is
  scanned for programs that are in the ExecutivePrefs Magic Wand list
  and these tasks will automatically be added to the preferences file.

* Ps now has a new built-in format string called STACK which displays
  stack usage statistics for each task.

```

#### V1.20

```

~~~~~
* Meter is a new client which displays all kinds of system information
  in a small window. It very customizable and you no longer need ten
  different programs to show all the information you need. See
  documentation for more information. Meter requires a registered
  version of Executive.

* Scheduler changes:

```

---

- o ALL SCHEDULERS
    - Nice-value now has much greater influence on task's priority.
  - o QUEUES
    - Reduced the number of queues from 10 to 8.
    - A task is not allowed to migrate upwards immediately, instead scheduler forces the task to wait for one second first. This eliminates priority "wobbling" which happens when two or more tasks using approximately the same amount of CPU time are fighting for the CPU.
    - This scheduler now adjusts Exec's Quantum-value when a task is launched. Lower priority tasks get higher Quantum, letting them run for longer time before switching to another task with same priority.
    - See documentation for more detailed information.
  - o SUPER
    - Remember task type (CPU intensive or interactive) for longer time
    - Task's type now influences task's priority even if task hasn't used any CPU time recently.
- \* Added a Flush memory menu item to top. This flushes unused libraries from memory.
- \* If a process has empty CLI command name (Term has if it's started from Workbench) Executive will now use task name instead when search for a task from watched tasks list.
- \* ALoad[\_MUI] can now display current CPU usage instead of system load average. You can also specify which colors should be used and you can optionally have a clock in the window's titlebar.
- \* Top can now be run in a shell-window, without opening an Intuition window, so you can run top from a remote shell. Use the STDOUT-option for this. There's also another new option, ONCE, which will just output the information to stdout and then exit.
- \* Commander and ExecutivePrefs windows are now much faster and take less memory.
- \* Commander now has a zoom-gadget and a new option, ZIP, for zipping the window when Commander is started.
- \* If a program was set to be [not] scheduled, and it was started using Run, it was possible that Executive didn't notice the task. This happens because the CLI command name is not set immediately when a task is created. And under heavy system load this may take several seconds. I added a workaround for this problem, but there's no bullet proof way to fix this completely. It should now work well enough in all imaginable cases.
- \* Added a fix for a bug in AmigaOS 3.x which causes system to lock up if a task that opens the Workbench screen has a priority lower than 1. Executive will now set task's priority to catchmax+1 while executing Intuition functions that could cause the Workbench screen to be reopened.
- \* Added some information to the Command problems section in docs.
-

- \* If the update interval was more than 1 second, ALoad[\_MUI] and ALoad3D displayed an average of the loadaverage, they now display the current value.
  - \* Stat will now show how many accounting records has been created. Stat will no longer output "Accounting is activated" because this should be obvious if accounting statistics are shown.
  - \* Install-script will now update existing preferences file if it finds tasks that are in the Magic Wand list.
  - \* If ExecutivePrefs[\_MUI] finds the preferences file from ENV: then preferences will be saved to ENV: AND ENVARC:.
  - \* Added a note about Delitracker MED-player to the Common problems section in docs.
  - \* You can now use CTRL-C to stop pstree.
  - \* Added MagicCX, MultiCX, ToolManager, TrapDoor, MagicMenu, MCP and JamMail to the Magic Wand list.
  - \* Uptime is now calculated from eclock instead of RAM:-disk creation time.
  - \* Programs started using WBStart-Handler are now accounted properly.
  - \* Added portuguese catalog.
  - \* MUI list headers are now shown with highlight pen, which is usually white.
  - \* Removed text from top documentation that said it's possible that top may cause serial transfer errors. This is not true.
  - \* You can now use TASK+CHILDTASKS option in the preferences file, instead of two separate options TASK and CHILDTASKS.
  - \* Fixed a small bug that caused a wrong item to be selected when double-clicking an item in ExecutivePrefs listviews.
  - \* Optimized the code that reads the preferences file, this made the server and ExecutivePrefs[\_MUI] a bit smaller.
  - \* Fixed a bug in ctp and kill-clients. If NAME, USER, GROUP or PGRP-options matched multiple tasks, it was possible that not all tasks were affected.
  - \* Registration fee is now accepted in DKK (Danish krone).
  - \* Money transfer to my bank account is now possible. See documentation or the registration forms.
  - \* ALoad and top now use pens from screen DrawInfo.
  - \* Top now has options for specifying background and text colors.
  - \* Added a zoom-gadget to the top-client. Pressing it causes the window to be zipped so that only the header is displayed. There's also a new
-



option, ZIP, for zipping the window when top is started.

- \* ALoad and uptime windows are now simple refresh windows, which take less memory than smart refresh windows.
  - \* Top will not allow a user (except the superuser) to break other users' tasks if MultiUser has been installed.
  - \* The clients that have a STAYFRONT option will now check if the window is frontmost before trying to bring the window to front.
  - \* Changed the ShapeShifter entry in Magic Wand list so that the ShapeShifter main task is scheduled, but not it's childtasks.
  - \* Corrected a lot of spelling errors in the docs. :) Also made some small improvements here and there.
  - \* Fixed a bug which caused some memory to be trashed when a client or the server displayed the AmigaGuide document.
  - \* Fixed a bug which caused task name to be displayed instead of CLI command name if process has a CLI structure but no CLI number.
  - \* Commander and ExecutivePrefs now retain listview position when updating the list. This only worked with OS3+ before.
  - \* Fixed the COMMAND-keyword in ps.
  - \* All icons now have tooltypes with default values.
  - \* Added some words to the Glossary in docs.
  - \* When top erased a task that became idle, it sometimes outputted a few miscellaneous characters.
  - \* Added Brilliance to the Magic Wand list.
  - \* When Commander's window is zipped, it's width is halved.
  - \* Added a new option ANGLE to ALoad3D. You can now specify the initial rotation angle. It's now also possible to rotate the bar-chart using left and right cursor keys.
  - \* Added a note about Term to Common problems section.
  - \* ALoad3D's display is now rendered immediately after the window is resized.
  - \* Commander will now update the task list when it's window is unzoomed. Commander\_MUI does the same thing when it's uniconified.
  - \* Top will no longer look for VMM every second if it isn't in use when top is started.
  - \* Added DirectoryOpus to the Magic Wand list. This is not necessary, but DO works faster and is more responsive when it's support tasks are not scheduled.
-

- \* ExecutivePrefs[\_MUI] now checks if the Executive.guide file is in BASEDIR. If not, it sets BASEDIR to the current PROGDIR: there when prefs are saved.
- \* The server now changes its priority to 126 after it has read the keyfile and preferences so no disk access is done at this priority.

### V1.30

- ~~~~~
- \* Removed some 68020+ optimizations from the server that were actually SLOWER than equivalent 68000 code.
  - \* Meter can now display number context switches during the last second.
  - \* Added Polish Install and UnInstall translations.
  - \* Registration price in Deutsche marks is now only 20 DM. It seems that 5DM notes are quite rare.
  - \* Discount in registration price is now available if three or more people register at the same time. See documentation.
  - \* The SPEED option in ALoad3D is now used to set the rotation speed in degrees. Default is 5\textdegree{}
  - \* The SPEED tooltip in ALoad3D now accepts value 0.
  - \* A new option BATCH in Top disables reading keyboard when information is displayed in a terminal. Another new option UIDS displays UIDs instead of user names.
  - \* A new option ALL in Kill sends all signals, C, D, E and F to a task.
  - \* GUI based clients now have a new menu item, "Project->Save config". This will save current settings to the client's icon, including window position.
  - \* Mutually exclusive menu items now work correctly.
  - \* Modified the installation script so that it now works even if you're using a program that patches the dos.library so that you get UNIX-like path handling. For example, Yak with the UNIX Dirs option turned on did this and caused the installation not to work correctly.
  - \* A new option MAX in Top displays as many tasks as possible.
  - \* When Top is running in a terminal, "n" can be used to change the number of tasks that are displayed.
  - \* ALoad3D now displays load averages more like ALoad. For load average 1.0 the bar is as high as possible. When the load average grows, the bar is divided in two parts. When the load average is for example 3.5, the bar is divided in four parts. Use the GROW option to get the old style bars.
-

- \* Fixed a nasty bug in the AddTask() patch. A Commodore debugging tool TNT didn't work if it was run before Executive. This bug might have caused crashes with other programs too.
  - \* Executive now has a nice logo, you'll see it in the server and ExecutivePrefs icons. The logo is available in different formats, in the Executive WWW-page.
  - \* A sanity check is now done to the task names. If a non-printable character is found, it's converted to '?'.  
\* Meter can now display a graph of a value, for example CPU usage. The new options GRAPHHEIGHT, GRAPHS, GRAPHCOLOR and GRIDCOLOR are used to control the graphs.
  - \* Added some preliminary MUI V3 support:
    - A menu item "MUI Preferences..." which opens the MUI preferences editor.
    - Increased the size of some string gadgets, MUI V3 seems to make them a bit too small.
    - ALoad\_MUI can use the Levelmeter-class in MUI V3 to display a nice car speedometer like meter of current CPU usage. This looks really neat, and is very usable too. The meter is currently updated four times a second.
  - \* Miscellaneous internal changes and some polishing here and there.
  - \* The new Extra-drawer contains the full-size Executive logo and a screenshot of the Meter-client.
  - \* Fixed the Register Arexx-script.
  - \* A new option NOTITLEBAR in ALoad, ALoad3D, Meter and Top disables the window titlebar.
  - \* Updated the Portugues and Nederlands catalogs.
  - \* Added German catalog and Install/UnInstall translation.
  - \* Added French catalog and Install/UnInstall translation.
  - \* Some SysInfo.library modifications:  
(Thanks to Bernhard Möllemann <zza@rz.uni-karlsruhe.de>)
    - Better names for some structs:
      - sysinfo -> SysInfo
      - loadaverage -> SI\_LoadAverage
      - loadaverage\_fixed -> SI\_LoadAverageFixed
      - SysInfo\_notify -> SI\_Notify
      - cpu\_usage -> SI\_CpuUsage
      - task\_cpu\_usage -> SI\_TaskCpuUsage
    - SysInfo was sometimes spelled as sysinfo or Sysinfo. It's now always SysInfo.
    - sysinfo.fd is now SysInfo\_lib.fd
    - Changed the compiler dependant types:
      - int -> LONG
      - short -> WORD
      - long -> LONG
-

- Removed LOADAVG\_FLOAT.
- Replaced the 'use\_messages' parameter in AddNotify() call with 'flags'.
- Some internal changes to support Executive V1.30

## V2.00

- ~~~~~
- \* All schedulers have been significantly improved. The basic principle behind each scheduler is the same, but they have been fine tuned and their specific characteristics emphasized. Nice-value has now more effect and focused task gets more CPU time than before.
  - \* Catalog translator's name is now displayed in the About... menu and when a program is started with the VERSION option.
  - \* Added a real multi-level feedback queue scheduler, similar to what is used in most UNIX operating systems, for example in all BSD variants (NetBSD, FreeBSD, BSDI, etc.). The new scheduler is called "FEEDBACK". See documentation for technical explanation.
  - \* The 68020+ version of Executive used a CMP2 assembly command which isn't supported by 68060.
  - \* Added NewIcons, thanks to Philip A. Vedovatti for drawing them!
  - \* When Meter was started with the NOTITLEBAR option, it stopped updating when a new value was selected from the Show-menu.
  - \* Dashboard is a new client which lets you open windows on public screens and create any number of all kinds of meters (CPU usage, free memory, SANA-II, VMM, Device I/O, etc.) on each window. Everything is configurable.
  - \* ALoad, ALoad3D, Meter and Top now have invisible drag-, close-, depth- and size-gadgets if the system gadgets have been disabled with NOTITLEBAR or NOSIZEGADGET option.
  - \* Added a Scan-button to the Tasks-section of ExecutivePrefs. It will scan the system for tasks in the Magic Wand list.
  - \* Added Amiga-Oberon interface for SysInfo.library. Thanks to Thomas Igracki, <T.Igracki@bamp.berlinet.de>.
  - \* Error requester windows are now opened on the screen where the client's window is, not on default public screen.
  - \* Fixed a bug that caused an Enforcer hit when Top was started with the STDOUT option.
  - \* Removed some unnecessary GUI-generation code. Timer and the GadTools versions of Commander and ExecutivePrefs are now a bit smaller.
  - \* GadTools versions Commander, ExecutivePrefs and Timer now use the screen font for the GUI, not the system default font, which is always non-proportional. The system default font is still used for listviews.
  - \* Added a checkbox to ExecutivePrefs[\_MUI]'s task-edit window, which is
-

used to choose if task's nice-value should be set to the given value or if parent task's nice-value is used.

- \* MUI V3 is now required by all MUI-based clients.
  - \* Added Croatian (Hrvatski) catalog.
  - \* It's no longer necessary to give the JOINLONELY option with THRESHOLD in Sa, as it will be set automatically.
  - \* Sa displayed "NaN0.00%" for CPUTIME% if total CPU time was 0.
  - \* Executive server and Acct now support PatchControl.
  - \* Dashboard, Meter and Top use "vmem.library" to get virtual memory statistics from GigaMem. It seems that there are other libraries with the same name, and these of course aren't compatible. By default these clients don't try to open the library, unless the VMEM option is given.
  - \* Added JUNK and FORCE options to Sa. These are used to strip garbage from the summary files.
  - \* Fixed a bug in Meter which caused values to disappear if the window didn't fit on the screen after "Show graphs" was selected.
  - \* If a new task is added in ExecutivePrefs and the task name is too long, the name will be cut and an asterisk "\*" added to the end.
  - \* Fixed a refresh error in ALoad.
  - \* When the server or a client is started from CLI, errors will now be shown in a requester if the output is directed to NIL:.
  - \* Meter didn't save device names into SHOW-tooltype with the "Save config"-command.
  - \* Meter now adds volume names to the Diskspace-submenu, as well as devices. Volume names can also be used with the SHOW option. New options NOVOLUMES and NODEVICES are used to disable volumes or devices from the Diskspace-submenu.
  - \* If Timer couldn't create an app-window, one signal bit wasn't freed.
  - \* ALoad, ALoad3D, Commander, ExecutivePrefs, Meter, Timer, Top and Uptime now open the AmigaGuide document asynchronously when Help-key is pressed or when "Help" is selected from menu.
  - \* Executive can now use the power LED to indicate when a task is running. The new option, LED, enables this feature.
  - \* The three bars in ALoad3D are now of different size. The thickest bar is the 1 minute load average and the thinnest bar represents 15 minute load average.
  - \* Fixed ExecutivePrefs[\_MUI] filerequester. When a preferences file is opened with the Open... menu, the file name will be correct in
-

the Save as... file requester.

- \* Reduced memory usage of the server and all clients by almost 4KB by removing one `atexit()`-command from the code.
  - \* Uptime's window is now a bit wider, it should be of correct width now.
  - \* Ps now has a new option, `STATUS`, which displays the same CLI processes as the AmigaDOS Status command.
  - \* ExecutivePrefs now warns the user about the `IGNORE`-type, as it really shouldn't be used. It's there for a very few buggy programs only. `NOSCHEDULE` should be used instead.
  - \* Lots of changes in Meter:
    - Graphs can be individually turned on/off by double clicking the value.
    - The `SHOW` command line option or `tooltpe` can be used to specify if a graph should be displayed for a value. Just add a "-" after the value name. For example: `SHOW=fast,total-,cpuusage-,uptime`
    - If labels, numbers and bars are turned off, only a graph is displayed.
    - The window can now be drawn without the standard window border (`BORDERLESS`) and put behind all other windows (`BACKDROP`).
    - Frames can be drawn for the window (`WINDOWFRAME`) and for individual values (`VALUEFRAMES`).
    - Space between window frame and value frames can be set using `SPACEHOR` and `SPACEVER` options, and space between the values can be set using the `BETWEEN` option.
  - \* `ALoad` (not `ALoad_MUI`), `ALoad3D`, `Top` and `Uptime` are now commodities and support Stefan Becker "`screennotify.library`".
  - \* `ALoad`, `ALoad3D`, `Meter`, `Top` and `Uptime` now open a window on the default public screen, if the specified public screen is not found.
  - \* If `ALoad`'s window was resized before it could redraw itself after an earlier resize, size after the first resize was used.
  - \* It's now possible to account ALL tasks with `Acct`, not just CLI commands or programs started from `Workbench`. This is enabled with the new `ALL` option in `Acct`.
  - \* If signals were used for notification in `SysInfo`, `RemoveNotify()` crashed when it tried to reply to unused notification messages in the message port, which isn't allocated if signals are used.
  - \* `Acct` now supports all versions of `WBStart`. Programs started with it are properly accounted.
  - \* By default, the maximum priority for a task to be caught is now +1, it used to be +2. So tasks with priority +2 or higher are not scheduled automatically. Dynamic range is now from -70 to -50, this lets tasks which use approximately the same amount of CPU time to be round-robin with the same priority.
  - \* The `SIMPLE` scheduler was TOO simple, so I removed it. If you're currently using this scheduler, please use the preferences program to change the scheduler. The `install-script` automatically changes
-

the SIMPLE-scheduler to FEEDBACK during installation.

- \* Added a new "QUEUE" keyword to the Ps-client. When QUEUES or FEEDBACK scheduler is used, this shows task's run queue.
  - \* It's now possible to schedule a group of tasks so that their relative priorities are maintained. This is done with the new RELATIVE option, that is used together with CHILDTASKS. There are some applications that use multiple tasks which are expected to have certain priorities. Please read ExecutivePrefs documentation on how to use this option.
  - \* Added ExecutiveAPI, a simple, easy to use application programmers interface to some Executive features. The most important feature is the ability to control task scheduling without having to add entries to Executive.prefs file.
  - \* Meter and Uptime no longer disable interrupts while calculating number of users logged in (MultiUser). This caused hardware overflow errors in serial transfer.
  - \* Disk usage bar in Meter displayed free space available on disk, now it displays how much disk space is in use, which is more logical.
  - \* If SysInfo.library exists in LIBS:, it will be updated. Otherwise user will be asked if the library should be installed.
  - \* Added "About MUI..." menu item to all MUI-based clients.
  - \* Fixed some backgrounds in Commander\_MUI and ExecutivePrefs\_MUI, they are now user configurable.
  - \* Improved/redraw most of the icons.
  - \* New option in Top, CURRENT, displays current CPU usage instead of recent CPU usage.
  - \* Top's window now has a size gadget, if the SIZEGADGET option is given.
  - \* New option, SCALESTEP, in ALoad[\_MUI] controls which scalelines are drawn. If SCALESTEP is 3, every third scaleline is drawn.
  - \* ALoad[\_MUI], ALoad3D and Top now have a BACKDROP option which makes the window a backdrop window that is always behind all other windows and can't be depth arranged.
  - \* Executive is now a commodity. Scheduling can be disabled/enabled using a commodities exchange program. Stat-client displays if scheduling is disabled or enabled.
  - \* Lots of changes in Magic Wands:
    - ADDED: FinalWriter, XiPaint, MagiC64, BlitzBlank, DeluxeMusic, DreamTerm, SwazBlanker, MultiCX Blanker, Garshneblanker, Termite.
    - REMOVED: Blanker, Flying Toasters!, Midnight, Rotor.
    - CHANGED: BServer - Schedule blankers with nice-value +20.  
Garshneblanker - Schedule blankers with nice-value +20.
-

DesktopMAGIC - Schedule the blanker task with nice-value +20.  
Brilliance - Recognize TrueBrilliance also.  
MultiCX - The correct task name is "MultiCX".  
MCP - Recognize MCP020 also.  
WBStart Handler - Recognize V2.x also.

- \* Added labels to Magic Wand listview.
- \* Added a small interactive tutorial to the AmigaGuide documentation. It takes you through some Executive features and clients. There's now a small "Boxes" program in the Extra-drawer, which is used by the tutorial.
- \* Updated the Meter screenshot.
- \* The documentation has been enhanced, lots of new information has been added.
- \* There are some applications that support Executive through SysInfo.library, added a list these to the Clients-section of the documention.
- \* Price drop from 120 SEK -> 100 SEK (Swedish krona), other currencies unchanged.
- \* Australian (20 AUD) and Canadian (20 CAD) dollars are now accepted.
- \* Postal Money Order is now accepted.
- \* Executive can now be properly installed on systems that are running a utility which converts standard Amiga pathnames to UNIX pathnames, i.e. "/" becomes ":". Normally the "/" refers to parent directory.
- \* The install-script now asks the user if he's using some of the applications which require a special configuration for Executive. The selected applications are automatically added to Executive preferences.
- \* Executive now defaults to using timer.device instead of CIA-B timer on 68020+ systems. The install-script will change CIA-B timer to timer.device (which uses CIA-B).
- \* Numerous small bug fixes and improvements.

---

Changes made after V2.00beta:

- \* Dashboard no longer asks for confirmation when meters are cut.
  - \* Fixed a bug in Dashboard: if load average was higher than 1.0, then the first line for a load average graph was too high. The display wasn't divided.
  - \* The Install-script didn't work with new versions of Installer, English was always used and no catalog was installed.
-



- \* Fixed a small vertical alignment bug in Dashboard's text-type meters.
- \* Hopefully Executive now works correctly with PatchControl.
- \* Dashboard and Top now work properly with sysihack and similar programs.
- \* Fixed a CHIP memory leak in Dashboard's bar-type meter.
- \* Fixed a small memory trashing bug in ExecutivePrefs. This occurred during first installation of Executive, when initial preferences file was written.
- \* Because of an odd-address error Dashboard couldn't save the preferences when running on 68000.
- \* Stat displayed "Total used CPU time" instead of "Total idle CPU time".
- \* Now clients that have a window will always display error messages on windows, other clients display errors on windows only when started from Workbench.
- \* There's no default SANA-II device name on Dashboard anymore.
- \* If Dashboard couldn't remove its DoIO() and SendIO() patches when quitting, ScreenNotify was turned on and screens couldn't be opened or closed anymore.
- \* Dashboard's round-type meters now work on a CyberGraphX display.
- \* If Meter used borderless window, the size was calculated incorrectly.
- \* Added Norwegian and Italian translation.

## 1.6 Introduction

### INTRODUCTION

~~~~~

Getting started

Multitasking basics

Different kinds of tasks

Scheduling

Executive features and requirements

How does it work?

Installed files

Moving Executive to another directory

---

If you want to be informed when a new version of Executive is released, please send your email address to me . Registered users are automatically added to the mailing list. ←

## 1.7 Getting started

### GETTING STARTED

~~~~~

If you have installed Executive normally and rebooted your system, it's now running and you can proceed. If not, go ahead and install it, Executive can be easily and safely uninstalled using the uninstall script.

Executive consists of a server and clients . The server is run in the background and most of the clients require it to be running to work.

In a few words, Executive adds a better task scheduler to your Amiga and provides similar services as programs like Job Manager or SPY system.

No similar programs for Amiga have been made before, so you may not exactly know what Executive is all about, but give it a little time and read the documentation and you won't regret it!

Please at least read the Common problems section.

### UNREGISTERED VERSION

These features require a registered version of Executive:

- \* All schedulers except STANDARD
- \* Accounting
- \* ALoad3D
- \* Meter

The absence of these features doesn't make Executive unusable. You can use Executive without these features but I hope you'd register after you have seen what this program is.

### BUGS

---

If you find a bug or have a suggestion for improving Executive, please contact me.

If you want to report a bug, please include the following information:

- Your Amiga hardware (Amiga type, CPU, memory)
- Your AmigaOS operation system version
- Other relevant software running on your system
- Executive version

If possible, try to find the necessary steps to reproduce the bug.

## 1.8 Multitasking basics

### MULTITASKING BASICS

~~~~~

Multitasking is something that Amiga users have been enjoying for 10 years while others a few years ago were still arguing if multitasking is a useful feature in an operating system. You are probably familiar with multitasking, but I'd like to clarify some terms and draw an overall picture of different ways to implement multitasking on a uniprocessor system. This will help you understand what Executive has to offer for you.

### MULTITASKING ON AMIGA

Amiga's multitasking system is called preemptive prioritized Round-Robin. Each task has a priority assigned to it. The task that has the highest priority and is ready to run, will be run. A task can be in one of three states: sleeping, ready or running.

- \* A ready task is waiting to use the CPU. Exec keeps a sorted list of tasks in ready-state. The list is sorted according to task priority, so Exec can easily find a task with the highest priority.
- \* A sleeping task is waiting for some event to happen. When the event occurs, Exec will move the sleeping task into the list of ready tasks.
- \* A running task is currently using the CPU. It will remain the current task until one of these three things occur:
  - o A higher priority task becomes ready, so Exec preempts the current task and switches to the higher priority task. The current task is moved to the top of the ready-list.
  - o The currently running task needs to wait for an event, so it goes to sleep and Exec switches to the highest priority task in Exec's ready list.
  - o The currently running task has had control of the CPU for at least a preset time period called a quantum and there is another task of equal priority ready to run. In this case, Exec will preempt the

current task for the ready one with the same priority. This is known as time-slicing or "Round-Robin". When there is a group of tasks of equal priority on the top of the ready list, Exec will cycle through them, letting each task use the CPU for a quantum.

The term preemptive prioritized Round-Robin comes from these three features.

#### PREEMPTIVE VS. CO-OPERATIVE

Preemptive multitasking means that the operating system can force a task to give the CPU to another task. Some operating systems have co-operative multitasking, mainly because they are old and their developers didn't see any need for better multitasking (or for any kind of multitasking) when the OS was developed. It's usually not possible to implement preemptive multitasking afterwards, so co-operative multitasking must be used.

In co-operative multitasking operating system tasks must handle the multitasking, i.e. do someone else's homework.

#### PRIORITIES

Amiga has fixed priorities. Most user tasks have a priority -1, 0 or 1. Tasks that require CPU time before user tasks, for example device drivers, have a priority 5. Harddisk partitions and disk drives have a priority 10 and the highest priority is usually 20, for input.device that handles all input from mouse and keyboard.

The priority range is -128 to +127. Negative priorities are usually used for tasks that require all the CPU time they can get, for example rendering programs. As a general rule, you shouldn't raise the priority of any normal task above 3.

Fixed priorities can cause starvation, a task doesn't get any CPU time at all before a high-priority task exits or starts to wait an event. It is rumoured that when MIT shut down its 7094 in 1973, there was still a low-priority job from 1967 on the machine.

## 1.9 Different kinds of tasks

#### DIFFERENT KINDS OF TASKS

~~~~~

Tasks can be divided in to two categories:

- \* transput-bound
- \* compute-bound

Transput-bound tasks spend most of their time waiting for some event to occur. Transput-bound tasks can be interactive or non-interactive. An example of an interactive transput-bound task is a text editor, it spends most of the time sleeping, waiting for a keypress. Device drivers

---

are non-interactive transput-bound tasks.

Compute-bound tasks spend their time using as much CPU time as they can get. Compute-bound task can also be interactive. An example of this kind of task is a word-processor, when it's printing a document, or a rendering program when its calculating an image. Most compute-bound tasks are non-interactive.

## 1.10 Scheduling

### SCHEDULING

~~~~~

The goal of scheduling is to provide good service to all tasks that are currently competing for the computing resource, that is, the execution of instructions.

#### DIFFERENT TASKS HAVE DIFFERENT NEEDS

A personal computer like Amiga usually has non-interactive transput-bound tasks, interactive transput-bound tasks and non-interactive compute-bound tasks. Different kinds of tasks must be given different treatment:

- \* consistent response time for interactive transput-bound tasks
- \* small response time for non-interactive transput-bound tasks
- \* good throughput for compute-bound tasks

Interactive transput-bound tasks require immediate response to every event, otherwise the user might get very frustrated. Failing that, consistent response is better than good average response. Response to a keypress that always takes about 2 seconds is better than response that averages about 1 second but occasionally takes 10 or only 0.5.

Non-interactive transput-bound tasks require fast response to events, because otherwise they'll be delayed. For example, consider a task that needs to compute for 1 millisecond and then wait 20 milliseconds for an event to occur. This will be repeated 1000 times. This would require 1 second of computation and 20 seconds of waiting, a total of 21 seconds. But if the task is delayed half a second every time it becomes ready, it will take 521 seconds. A small average delay will pay off handsomely, even if some delays are longer. A 1-millisecond average delay will allow the task to finish in 22 seconds.

Compute-bound tasks require lots of CPU time, and the overhead should be minimized, i.e. the number of task switches should be minimized. A context switch on AmigaOS is much quicker than in some larger operating systems. Here are some context switch speeds for different machines/operating systems (in microseconds):

26	25MHz Amiga A4000/040 AmigaOS 3.1
71	100MHz Pentium Linux 1.2.13
89	233MHz AlphaStation 255/233 Digital Unix 3.2
102	150MHz DEC 3000/300 OSF/1 2.0
106	66MHz Snake HP-UX 9.x

126	25MHz Amiga A2000/030 AmigaOS 2.1
128	40MHz Viking SunOS 4.1.3
131	167MHz SPARC Ultra-1 Solaris 2.5
210	33MHz 486 386BSD 0.1
212	50MHz RIOS AIX 3.2

#### HOW TO DISTINGUISH DIFFERENT KINDS OF TASKS?

This is a bit technical and you may want to skip it.

To make scheduling decisions, the scheduler must first distinguish the category where a certain task belongs. This can be very difficult because a task can drift from one category to another while it's running.

For example, a text editor is usually interactive transput-bound task, but when user starts a search & replace-function, it might become interactive compute-bound task. When it's saving text to a disk, it becomes non-interactive transput-bound. This means that the scheduler must adapt to different situations while the task is running. The best scheduler would be the one that could predict the future.

When implementing a scheduler the usual approach is to give priorities to each task and then modify these while the task is running. The simplest form of this is a decreasing priority scheduler where the priority of a task decreases continuously as it uses processor time, and the scheduler runs higher priority tasks first. Operating system called Multics used such a scheduler and discovered its major disadvantage, that on a heavily loaded system with significant numbers of short-lived tasks the priority of a lengthy task can decrease to a point where little or no further processor time is available to it.

To avoid the problem of permanently depressed priorities it is necessary to elevate them in some manner. There are two major choices for elevation methodologies: event-based elevation and processor usage aging. Event-based elevation is used to deliberately favor interactive response over compute-bound tasks because task priority is elevated by the occurrence of events such as keypress. This methodology assumes that tasks are either distinctly compute-bound or distinctly interactive (transput-bound) and that interactive tasks are more important. Tasks whose interactive work consumes large amounts of processor time may not do well under this methodology. The VAX/VMS scheduler employs this elevation methodology.

The second priority elevation methodology is processor usage aging. A scheduler that uses this methodology elevates priorities by gradually forgetting about past processor usage, usually in an exponential fashion. As a result, the scheduler's measure of processor usage is an exponentially weighted average over the lifetime of a task. A simple exponential average is not desirable because it has the unexpected side effect of raising priorities when the load on the system rises. This happens because under higher load each task gets a smaller share of the processor, so its usage average drops, causing its priority to rise. These elevated priorities can degrade system response under heavy loads because no task accumulates enough usage to drop its

priority. The 4.3BSD version of UNIX solves this problem by making the aging rate depend on the load average, so that aging is slower in the presence of higher load, keeping priorities in approximately the same range.

#### DOES AMIGA NEED A BETTER SCHEDULER?

A better scheduler would be useful. It would free you from constantly manually adjusting task priorities.

There are many situations where Amiga slows down so much that it basically becomes a single-task system. For example, you might be printing a large document from a wordprocessor and it would take too much time to load another program so you'll just wait for the word processor to finish. Even starting a task manager to change the word processor's priority can take 10-20 seconds!

When you start a time consuming task, you have to manually lower its priority before or after so that you can run other programs. For a "power user" this soon becomes a nuisance.

It's also easy to lock up the system by raising the priority of a task that wants to use all processor time, above other tasks.

I have also run into problems with text editors. Their priority is usually 1 or 2, which is higher than the default priority 0. When you load a large text-file to a text-editor and start a time consuming operation, for example macro execution or a vertical cut in CygnusEd, you can't do anything else before the operation finishes.

A better scheduler would handle situations like these automatically, without any user interaction.

## 1.11 Executive features and requirements

### EXECUTIVE FEATURES AND REQUIREMENTS

~~~~~

Executive is:

- \* a scheduler
- \* a task management system

Executive is fully compatible with all Amiga computers running AmigaOS release 2.04 or newer.

Executive is a client-server application. The server is usually started when the system is booted. It will install the scheduler (optional) and collect statistics of CPU usage. Clients are used to query information from the server.

Here's a list of Executive's major features:

---

- \* six different schedulers
- \* focus - task with active window gets more CPU time
- \* accounting
- \* load averages
- \* precise (1/1000 seconds) CPU usage timing
- \* tracks child-parent task relationships
- \* process identifiers, process groups
- \*
- multiuser.library
- support
- \* SysInfo.library for user-implemented clients

Other important things:

- \* localized
- \* on-line help
- \* most GUI-based clients available with GadTools and MUI interfaces
- \* server and clients use memory pools, no memory fragmentation

Here's a list of client-programs included:

|           |                                             |
|-----------|---------------------------------------------|
| Acct      | Accounting daemon                           |
| ALoad     | Display load average or CPU usage           |
| ALoad3D   | Display load averages in 3D                 |
| Commander | General task manager                        |
| Ctp       | Extended ChangeTaskPri                      |
| Dashboard | The mother of all meters                    |
| Kill      | Extended Break                              |
| Lastcomm  | List last commands executed                 |
| Meter     | Display system information                  |
| Nice      | Run programs with lower scheduling priority |
| Ps        | Process status displayer                    |
| Pstree    | Display child-parent relationships          |

---



```

Renice
    Renice a task

Sa
    Display accounting statistics

Stat
    Display some misc. information

Timer
    Time shell and Workbench programs

Top
    Display information about the top CPU tasks

Uptime
    Display system uptime and load averages

```

Will those clients run from a shell only?

A student asked the master for help... does this program run from the Workbench? The master grabbed the mouse and pointed to an icon. "What is this?" he asked. The student replied "That's the mouse". The master pressed control-Amiga-Amiga and hit the student on the head with the Amiga ROM Kernel Manual.

-- Amiga Zen Master Peter da Silva

## 1.12 How does it work?

HOW DOES IT WORK?

~~~~~

The first thing that must be done is to start the server. Most of the clients require that the server is running because they need to get information from it. Server should be started as early as possible, preferably in startup-sequence, but it can be started and stopped later. If you move Executive from "user-startup" to "startup-sequence", make sure it's started after SetPatch and Emplant's

```

RsrvMem
. Please don't

```

try to put Executive server to virtual memory. Executive is basically a part of the OS and needs to access certain data structures during times when its impossible to fetch them from disk, if they have been swapped out.

When the server starts, it reads the preferences file. Most of the options can also be specified in tooltypes and on command line. You can modify the preferences file with the

```

ExecutivePrefs
preferences

```

program.

The server will patch

---

some functions  
in exec and intuition libraries.

It necessary to run the server at a very high priority, currently at 126, otherwise the CPU usage calculations would not be accurate. This won't cause any problems with other tasks because Executive needs the CPU only for a very brief moment.

Executive doesn't replace Exec task dispatching functions, it just recalculates task priorities for scheduled tasks every second.

Two priority ranges are needed. The first one is called "catch range" and the other is "dynamic range".

#### Catch range

Tasks that have priority inside the catch range will be automatically scheduled. The default range is from -105 to +1.

#### Dynamic range

When the scheduler calculates a new priority for a task, it will be in this range. The default range is from -70 to -50. Task that uses little or no CPU time, will get priority close to -50. Tasks that use much CPU time will have priority near -70.

So, not all tasks are scheduled automatically. You can still run programs with priority above and below the catch range. For example, input.device will run at priority 20 as usual.

Different schedulers calculate the priority in a different way, but they all use a nice-value associated with each task. Nice-value is usually 0, but it can be used to give more or less CPU time to some tasks. The nice-value is in range -20 to +20. Unlike priorities on Amiga, a NEGATIVE value gives MORE CPU time to a task, i.e. it will be less nice to other tasks.

Another thing common to all schedulers is the focus-feature. The task that has opened the currently active window gets more CPU time than others. This feature can be turned off.

When using a scheduler you can specify certain tasks that will be watched and some action that will be taken when this task is created. This feature is needed when you don't want to schedule a task whose priority is in the catch range or you especially want to schedule some task that has priority outside the catch range. It's also possible to give default nice-values or priorities for tasks when they are created.

The main use for this is to make sure that some programs like telecommunication programs are not scheduled, because that might drop their priority too low and cause hardware overflow errors in serial transfer.

The watched tasks feature is also used to specify certain tasks that should be completely ignored. There's currently only ONE task that must be ignored. The SAS/C Development System has a debugger called CPR which creates a task named "VISOR input processor". This task is not removed in a legal way (RemTask() is not called), so it will cause problems with

programs like Executive. But because Executive handles this task AUTOMATICALLY, you don't have to worry about it.

There's absolute no need to ignore any other tasks.

When the server is running, you can start some clients, for example Top and ALoad. While a client is running, you can't quit the server.

## 1.13 Patched functions

### PATCHED FUNCTIONS

~~~~~

exec.library

AddTask()  
RemTask()  
SetTaskPri()  
Switch()

intuition.library

CloseWindow() - when FOCUS feature is ON  
OpenWindow() - when FOCUS feature or OpenWorkbench fix is ON  
OpenWindowTaglist() - when FOCUS feature or OpenWorkbench fix is ON  
OpenWorkBench() - when OpenWorkbench fix is ON  
LockPubScreen() - when OpenWorkbench fix is ON  
CloseScreen() - when OpenWorkbench fix is ON

ExecBase

ex\_LaunchPoint

## 1.14 Installed files

### INSTALLED FILES

~~~~~

All Executive files, except two locale catalogs are installed to a single directory. The preferences file, "Executive.prefs" is initially written to this directory, but Executive will also search it from these directories:

current directory  
PROGDIR: (directory where the program binary is)  
ENV:  
S:

These directories are also searched for the  
keyfile  
, "Executive.key".

A preferences file containing all the defaults is created during installation. It also contains the BASEDIR option, for example:

```
BASEDIR      "WORK:Executive"
```

This is the directory where Executive was installed. The AmigaGuide document file is searched from this directory when you start a client from a shell with the HELP option or press the HELP-key when using a client with a GUI. These directories are also searched for the documentation file:

```
PROGDIR:Help/<language>/
HELP:<language>/          (AmigaOS V39+)
PROGDIR:
Current directory
```

Two locale catalogs are installed if you are using AmigaOS 2.1 or newer. AmigaOS 2.04 doesn't support localization. The catalogs (Executive.catalog and Dashboard.catalog) are installed to LOCALE:Catalogs/<language>/.

To uninstall Executive click on the UnInstall icon and follow given instructions. This will safely remove Executive from your harddisk.

## 1.15 Moving Executive to another directory

### MOVING EXECUTIVE TO ANOTHER DIRECTORY

Experienced users may want to move Executive to another directory. Here's how to do it:

Copy the server and clients to a directory that is in the path, for example C:.

Move the DashboardPrefs directory somewhere, and use the PREFSDIR-option with Dashboard to specify where this directory is.

Copy the keyfile and the preferences file to your S: directory.

Copy the AmigaGuide manual to HELP:<your default language>/ directory.

Remove the BASEDIR line from the preferences file.

Remove the Path-command from S:User-Startup file and change the path to the Executive binary.

If you don't use Workbench, you can delete all the icons.

Of course the supplied UnInstall script can't be used anymore, so you can delete it.

## 1.16 Common problems

### COMMON PROBLEMS

~~~~~

Here are solutions for a few common problems that you might encounter when using Executive. These are not bugs, but "features" of Executive.

If you have any questions about Executive, please ask me, my email address is <petrin@megabaud.fi>.

### MAGIC WANDS

There are a few programs that shouldn't be scheduled.

ExecutivePrefs

has preset settings for these programs, you only need to copy ←  
them from

the Magic Wand list.

When you installed Executive your system was scanned for programs that are in the Magic Wand list and they were automatically added to the preferences file. This only works for programs that were running when you installed Executive, others must be added manually.

The Scan-button in the Tasks-section of ExecutivePrefs scans the system for tasks in the Magic Wand list and adds them automatically.

### ERRORS IN SERIAL TRANSFER

If a communication program is scheduled by Executive, its priority will drop when it CPU time, and then higher priority tasks may block it for a while. Due to limitations of Amiga serial port hardware, this will cause hardware overflow errors. The serial port can only hold a total of one byte of incoming data (plus one byte currently being transferred). The port requires CPU intervention to handle flow control, so any delay to the CPU's ability to respond to interrupts may - at higher baud rates - cause loss of data while receiving.

This problem is easily solved. Start

ExecutivePrefs

and go to the tasks-

section. Press the New gadget to create a new entry for the communication program. Press the Magic Wand gadget and see if the communication program you're using is on the list. If it is, just select it and press Ok. Press the Ok gadget in the Edit-window and save the preferences. Changes in preferences take effect after the server has been restarted.

Note that you have to start the communication program from Workbench, or use the Run-command in a shell, if the communication program doesn't automatically detach itself. If you just execute the program in your current shell, no new task is created and Executive won't notice the program at all.

If the communication program that you're using isn't on the Magic Wand-

list, then choose "Communication program" from the list and press Ok. Start the communication program so you'll see what's the name of it's task. Press the Name gadget (in the MUI version of ExecutivePrefs this is a small popup gadget) and select the communication program from the list, press Ok and save the preferences.

It's possible that the communication program creates several tasks. In this case all task names usually begin with the same string, for example "Term", so type "Term\*" in the name gadget.

Leave all other settings untouched (i.e. TASK, NOSCHEDULE and ABOVE selections should be activated).

#### SCREENBLANKERS

Screenblankers that use a lot of CPU time when the screen is blanked, shouldn't be scheduled or they will steal CPU time from other programs. Or, you can configure Executive to schedule them with nice-value +20, in which case they won't get as much CPU time as tasks with higher nice-values. ExecutivePrefs has Magic Wands for most commonly used screenblankers and also for MultiCX's internal blanker.

#### MULTI-FUNCTION COMMODITIES

Some multi-function commodity programs like MultiCX and MCP have features (e.g. SunMouse) which won't work properly if these programs don't get CPU time almost immediately when they need it. Although the scheduler keeps these tasks running at a very high priority (because they don't use much CPU time), it's still possible that they become blocked by some higher priority task. Executive should be configured not to schedule these programs.

#### SHAPESHIFTER

ShapeShifter, the Macintosh emulator requires that its main task's priority is kept below its childtasks' priorities. The Magic Wand in ExecutivePrefs configures Executive so that the main ShapeShifter-task, which uses all available CPU time, is scheduled and the childtasks, which don't use much CPU time, aren't scheduled.

ShapeShifter requires that you use either timer.device or the CIA-B timer, because it can only use CIA-A.

#### RSRVMEM

To make Executive work with RsrvMem, a program that is used with Emplant, you have to start Executive AFTER RsrvMem. RsrvMem REPLACES exec.library's Switch()-vector, instead of patching it and this also disables the patch that Executive has applied. Executive still runs, but CPU usage will not be calculated and all CPU usage values displayed by Top, Ps and other clients will be 0. If you start Executive after RsrvMem, it will properly patch the Switch()-vector.

## TOOLSDAEMON

If ToolsDaemon is scheduled, it may become blocked by a higher priority task and then it can't respond as fast as it should. Configure Executive not to schedule ToolsDaemon. There's a Magic Wand for this.

## BRILLIANCE

Brilliance is programmed in such a way that changing its subtasks' priorities causes some mouse and keyboard input to be ignored. Use the Magic Wand to cure the problem.

## GARSHNEBLANKER

This screenblanker seems to lock up or crash when its childtask "GarshnePing" gets higher priority than the main task. This problem can be fixed by telling Executive to keep this task's priority at -128. There's a Magic Wand for this.

## FINAL WRITER

Final Writer from SoftWood Inc. has a bug in its printing code. When printing a document, Final Writer may crash the system, or the output may be messed up. This occurs when the "printer.device" task has higher priority than Final Writer's subtask "FW5\_Process.x". This can happen when Executive is running, but because it's nowhere guaranteed that "printer.device" will always be run at priority 0 or lower, this is a bug, not a "feature". SoftWood has known about this for 2-3 years, but it has not been fixed and my last email about the bug seems to have fallen on deaf ears.

There are two ways to make Final Writer printing work with Executive. The Magic Wand in ExecutivePrefs configures Executive so that the "printer.device" task's priority is kept below the FW subtasks. The other way is to configure Executive to keep the subtasks' priorities above "printer.device". This can be done with the following entry in Executive.prefs file:

```
TASK    FW*_Process.*    NOSCHEDULE    ABOVE
```

## BROWSER II

There's a minor problem with BrowserII, if your CATCHMAX-value is 2 or higher. Starting from Executive V2.00 the default CATCHMAX-value 1, so BrowserII will work correctly. Otherwise the destination directory's window is not properly updated after files have been copied. Configure Executive not to schedule BrowserII and its childtasks.

## MAGICMENU

This doesn't set its CLI command name when started from shell. If

---

MagicMenu is started from Workbench, Executive thinks the name is "MagicMenu", but when it's started from shell, the name is "BackGround\_Process". There are two Magic Wands for MagicMenu, one is used when the program is started from a shell and the other when its started from Workbench.

#### MAPLEV 1.0

MapleV 1.0's Plot() and Plot3d() functions don't work when Executive is running. Maple creates a plot-task and its priority must be lower than the main MapleV task's priority. This has been fixed in later versions of MapleV. If you're using MapleV 1.0, add this to your Executive.prefs file:

```
TASK    MapleV    NOSCHEDULE    ABOVE
```

#### MCP

If you're using a Garshneblanker module with MCP, configure Executive to keep the priority of a task called "GarshnePing" at -128. This can easily be done with the "Garshneblanker (GarshnePing)" Magic Wand in ExecutivePrefs.

#### CHOOSING THE RIGHT TIMER

After you have installed Executive, you might get a message from some program that CIA-A or CIA-B timer is not available. Executive can use three different

```
timers
:
CIA-A
CIA-B
timer.device
```

Executive defaults to CIA-A on 68000/010 Amigas and to timer.device on 68020+ Amigas. Timer.device uses CIA-B timer, which may cause serial transfer errors on slower Amigas (this has nothing to do with Executive), so CIA-A is the best choice for 68000/010 Amigas.

#### EXECPATCH

If you're having problems like Top displaying 0% CPU usage, you might be using a hack called ExecPatch. I don't know what other problems it may cause because it hangs on my system.

#### OCTAMEDPLAYER, DELITRACKER AND MULTIPLAYER

If you're using CIA-A timer with Executive, some versions of OctaMEDPlayer crash because they can't allocate a timer. Get the latest version of OctaMEDPlayer from Aminet, the file is mus/play/OctaMED-P.lha.



MultiPlayer's and Delitracker's MED-players crash if you use CIA-A timer with Executive. Use ExecutivePrefs to change the timer to CIA-B or timer.device, both will work fine.

#### QUADRACOMPOSER

This program doesn't know how to use CIA-B timer if CIA-A is allocated. Configure Executive to use CIA-B timer or timer.device instead of CIA-A.

#### AIR

AIR is an infrared remote controller program from Geodesic Design. It uses the CIA-B timer without allocating it from ciab.resource. If Executive is configured to use CIA-B, AIR will mess up with Executive timing and crash is likely to occur. You must configure Executive to use CIA-A timer or timer.device if you want to use AIR.

#### VINCI

This text viewer has a bug, it can't be run with priority -1 or less so it will crash under Executive.

#### SCALA

I'm not sure if ScalaMM works correctly with Executive. If you experience crashes or other strange behaviour, let me know. Configuring Executive not to schedule the tasks "ScalaMM" and "scalamm.gfx" may help.

#### AMITCP

AmiTCP doesn't require any special configuration for Executive.

## 1.17 Tutorial

#### TUTORIAL

~~~~~

This short tutorial demonstrates some Executive features and clients. This tutorial requires that Executive is running and a scheduler is in use. If you have just installed Executive and haven't rebooted yet, do it now, otherwise Executive won't be running.

---

A small program called "Boxes" is used in this demonstration. This program just draws filled rectangles to a window using 100% of available CPU time.

#### SCHEDULING

Amiga's multitasking system is called "preemptive prioritized round-robin". The problem is that the task priorities are fixed. When you start two tasks with different priorities, the task that has higher priority always gets to run first, and if it uses 100% available CPU time, the other task never gets a chance to run. This is called "starvation". Please start the two "Boxes" programs by clicking the links below:

[ Start this first: ]

Boxes - low priority

[ And then this one: ]

Boxes - high priority

As you can see, only the high priority task gets any CPU time. One task can take over the whole system. Lets say you print a document from your word processor. You can't do much else at the same time because all CPU time goes to the word processor. Of course you can manually lower the priority of the word processor task, but because it takes all available CPU time, starting a task manager may take 10-20 seconds!

This isn't so under UNIX, OS/2, Windows NT or even Windows 95! Let's see what happens with the two tasks when Executive is running. Again start these two programs (if the two programs started above are still running, quit them):

Boxes

Boxes

Now both tasks get the same amount of CPU time. It's possible that one of the tasks gets to run for one second and then the other gets a chance to run. This happens especially with the Standard-scheduler. Sometimes both tasks are running at the same time. This depends on the scheduler being used, for example the Queues-scheduler does a quite good job at keeping both tasks at the same priority, if they use the same amount of CPU time. The Feedback-scheduler is even better, as it can adjust tasks priority much quicker than every second like the other schedulers.

Default priorities are no longer important, as long as the program's priority is in the catch range (default -105 -- +1). Executive will schedule all tasks in this priority range. Tasks with higher or lower priority usually shouldn't be scheduled.

Let's see what happens if the other task uses much less CPU time. Now the first "Boxes" task draws three rectangles and then waits for 1/50 seconds. The second task doesn't wait at all, it uses all available CPU time. Before starting the second task, observe how

fast the first one draws the boxes:

```
Boxes - Wait
Boxes - No wait
```

As you can see, the first task gets all CPU time it needs. On my 25MHz 68030 it uses ~10% of CPU time so Executive gives it higher priority than to the other task which uses 90% of CPU time.

When Executive takes care of task scheduling, you can start printing a document from your word processor and then start your favorite web browser and it will work at full speed!

Executive boosts the priority of the task which has opened the currently active window. The idea is to give more CPU time to the application that the user is currently working with. Again, start these two programs:

```
Boxes
Boxes
```

See what happens when you activate one of the windows.

#### NICE-VALUE

Executive lets you control how much CPU time each task gets. Each task has a nice-value, which by default is parent-task's nice-value, usually it's 0. The highest nice-value (max CPU time) is -20 and lowest value (nicest) +20. You can see their effect by starting these two programs with nice-values 0 and -20:

```
Boxes - nice value 0
Boxes - nice value -20
```

You can adjust the nice-value easily with the Commander-client. Let the programs started above still run and start Commander:

```
Commander
```

Select one of the "Boxes"-tasks and press the Nice-gadget. You can now adjust task's nice-value with the slider. Try changing the nice-value of the second task from -20 to +20 and see what happens.

You can start programs from a shell and assign them a nice-value using the Nice-client. The Renice-client lets you change nice-values.

#### KILLING

You can easily stop most programs with Executive. The easiest way to do this is with the Commander-client. Start the a "Boxes" program and Commander:

```
Boxes
Commander
```

Select the "Boxes"-task in Commander and press the Break-gadget. Don't

---

worry about the signal, just press Ok and the task will be killed.

The Kill-client is an extension to the original Break command included with AmigaOS. In fact, you can replace Break with Kill, as Kill is fully compatible with Break. Break only understands CLI numbers. The Kill-client lets you specify the task(s) which should be killed by PID (CLI or task number), process group, task name, user or group. Once again, start the "Boxes" program:

```
Boxes
```

And then kill the program with the following command (just click here):

```
Kill NAME=Boxes
```

## STATISTICS

The Executive package contains various clients that you can use to obtain all kinds of statistics of your system.

ALoad displays system load average for the past 1 minute or current CPU usage:

```
ALoad - load average
ALoad - current CPU usage
```

Top shows a list of tasks sorted by their recent (or current) CPU usage:

```
Top
```

Uptime displays a clock, how long the system has been up, how many users have logged in, and load averages for 1, 5 and 15 minutes:

```
Uptime
```

The Ps-client outputs a list of tasks currently in system. It's a bit like the AmigaOS Status command, except that it can display approximately 110 different values of each task. Here are a few sample outputs:

```
Ps          - The default
Ps STATUS   - Like the AmigaOS Status command
Ps ACTIVE   - Tasks that have recently used CPU time
Ps SCHEDULED - Scheduled tasks
```

Pstree shows the process tree, task-childtasks relationships. You can see if a task has created childtasks:

```
Pstree
```

Stat outputs miscellaneous statistics:

```
Stat
```

Ok, that was it, although you've just scratched the surface. There are more clients and options to see and try. Thanks for reading.

---

## 1.18 Server

SERVER

~~~~~

Starting the server

Making it work

Options

Server internals

Commodity

Quitting the server

Start server

Stop server

## 1.19 Starting the server

STARTING THE SERVER

~~~~~

If you have installed Executive normally, the server will be started when your system is booted.

It's possible to start Executive anytime you want to, but then it's not possible to resolve existing child-parent relationships so the

Pstree  
-client can't display them.

Also, if you want the focus-feature to work properly with all windows, you should start Executive before any windows have been opened.

The server can be started from a shell or Workbench. It must be "run" from a shell, so use this command to start it:

```
run <NIL: >NIL: Executive
```

## 1.20 Making it work

MAKING IT WORK

~~~~~

PRIORITY RANGES

---

Every task in Amiga has a priority, from -128 to +127. The priority is fixed and is usually only changed by the user. Executive will change priorities automatically and give lower priority to tasks that want to use much CPU time and higher priority for tasks that use less CPU time. Executive reserves a range of priorities called dynamic range where it places the scheduled tasks. The default range, -50 to -70 is normally unused, but a different range can be specified.

The dynamic range MUST be inside another priority range, called catch range. By default, the catch range is from +1 to -105. Every task that is started, or whose priority is later changed to a value in this range, will be scheduled. If some task is being scheduled and its priority is changed to some value outside the catch range, it won't be scheduled any more.

This is a crude map of the priority ranges mentioned above:

```

+127 - +
+126 - | Executive server runs at this priority, +126.
      |
      |
      |
      |
      |
      |
      |
+1  - | <-----'
0  - + | By default, tasks with
      | | priority from +1 to -105
      | | will be "caught" by the
      | | server and they will be
-50 - | <--' Scheduled tasks will be | scheduled.
      | | assigned a priority from |
      | | this area, the dynamic range, |
-70 - | <--' by default from -50 to -70. |
      | |
-105 - | <-----'
      |
-128 - +

```

The maximum dynamic range and catch range is from -125 to +125. Dynamic max - dynamic min must be at least 2.

It's possible to schedule ALL tasks in your Amiga, i.e. set catch range to -125 - +125, but usually tasks with priority 4 or higher work more efficiently when they are not scheduled.

If you don't use any of the schedulers, then you don't have to care about any priority ranges.

Why it's necessary to run the server with so high priority? Is it dangerous? No, it isn't. The server needs to be run with a priority above all other tasks so it can always be run when it needs to, otherwise CPU usage calculation would not be correct. It's not advisable to run any other tasks with priority 126 or higher, as it will affect CPU usage calculations.

## WATCHED TASKS

Usually you want to schedule all tasks that have their priority in the catch range. But there are some programs that shouldn't be scheduled for certain reasons. For example, a telecommunication program (NComm, Terminus, Term, ...) shouldn't be scheduled because its priority may drop below some other time consuming tasks and then it wouldn't get all the CPU time that it needs. Because of the limitations of the Amiga's internal serial hardware, this may cause transfer errors. So you must run this task with a priority above the catch range, or tell Executive not to schedule this task, even if its priority is in catch range. Task's priority won't be changed, if it isn't in the dynamic range, where the scheduled tasks are. If task's priority is in the dynamic range, its priority will automatically be raised above the dynamic range.

If the telecommunication program has an option to set its priority, you can use it and set the initial priority above the catch range, so the program won't be scheduled. If no such option is available, then tell Executive to handle it (this will be explained later). If the telecommunication program is not started from Workbench or RUN from a shell, Executive won't notice it, because a new task won't be created.

So, Executive watches new tasks and checks if their name matches a task in its watched tasks list. If it does, some of these actions might be taken:

- \* Keep the priority of a task above or below the dynamic range and don't schedule the task.
- \* Schedule a task even if its priority is not in the catch range. You can also specify a default nice-value for this task.
- \* Don't schedule a task but set its priority to a specified value. The priority can be changed afterwards. You probably won't need this.

These actions can be applied to the specified task, its childtasks or both.

Sometimes when a process (not a task but process) is created (e.g. with RUN command from a shell), it's possible that the CLI command name is not available immediately upon invocation. Executive will wait for approximately 1.5 seconds before it checks if this process is in the watched tasks list, the name should then be correct. After 5 seconds has passed since the process was created, Executive will check the name again if necessary.

Task names are case insensitive, and you can use \* as a wildcard. For example, "\*.device" will match all task names that end in ".device". If the task is a CLI process, then the CLI command name will be used.

Workbench is a special task that shouldn't be scheduled, because it would make Workbench rather unresponsive. This is built-in, so you don't have to worry about it. It's still possible to schedule the

Workbench task, there's an option to do this.

VERY rarely it's necessary to completely ignore a task. Currently I know of only one task that must be ignored. The SAS/C Development System has a debugger called CPR which creates a task named "VISOR input processor". This task is not removed in a legal way (RemTask() is not called), so it will cause problems with programs like Executive. But this will be handled AUTOMATICALLY by Executive, so there's no need to worry about it.

There's absolute no need to ignore any other tasks at the moment, but it's possible. If you ignore a task, then Executive clients don't know anything about it. Always use the NOSCHEDULE option instead of IGNORE.

## 1.21 Server options

### SERVER OPTIONS

~~~~~

Before parsing command line options or tooltypes, the server reads a preferences file. The easiest way to change the preferences file is to use the

```

    ExecutivePrefs
    utility. If you want to edit preferences
file by hand then please read the
    preferences file
    documentation.
```

Command line options and tooltypes override options specified in the preferences file.

It's not possible to change any options on the fly, you must always restart the server.

If the server is started from Workbench, tooltypes will be read, otherwise CLI arguments are parsed. The server understands these CLI arguments and tooltypes:

#### CATCHMAX

```

Template: CATCHMAX/N/K
Tooltype: CATCHMAX
Default:  1
```

The highest priority a task can have to be caught by the server and scheduled.

#### CATCHMIN

```

Template: CATCHMIN/N/K
Tooltype: CATCHMIN
Default: -105
```

The lowest priority a task can have to be caught by the server and scheduled.

#### DYNAMICMAX



Template: DYNAMICMAX/N/K  
Tooltype: DYNAMICMAX  
Default: -50

The upper bound for dynamic range.

#### DYNAMICMIN

Template: DYNAMICMIN/N/K  
Tooltype: DYNAMICMIN  
Default: -70

The lower bound for dynamic range.

#### NOFOCUS

Template: NOFOCUS/S  
Tooltype: NOFOCUS  
Default: Focus option is ON by default

If you specify this switch, the focus-feature is disabled. By default, the task that owns the active window gets more CPU time.

#### SCHEDWB

Template: SCHEDWB/S  
Tooltype: SCHEDWB  
Default: Workbench is not scheduled by default

Workbench is not scheduled, because that would cause noticeable slowdown in performance. Use this switch to schedule Workbench.

#### DONTFIXOPENWB

Template: DONTFIXOPENWB/S  
Tooltype: DONTFIXOPENWB  
Default: Fix OpenWorkbench bug

There's a bug in AmigaOS 3.x which causes system to lock up if a task that opens the Workbench screen has a priority lower than 1. Executive fixes this bug by raising task's priority to catchmax+1 or to at least 1. Task's priority will be set back to normal after the buggy system function has been executed. You can disable this with this option.

There are some utilities like MultiCX and MCP that also know how to fix this bug. They set task's priority to 1, but if you are running Executive, this has no effect. If this priority is inside the catch range, Executive will schedule the task and its priority won't be changed. You should disable the fix OpenWorkbench bug feature from these utilities if you're using Executive.

#### LED

Template: LED/S

---

Tooltype: LED

Use the power LED to indicate CPU usage. When a task is running, the LED will be bright. When no task is running, the LED is turned off.

#### TIMER

Template: TIMER/K  
Tooltype: TIMER  
Default: CIA-A (on 68020+ Amigas timer.device [=DEVICE])

Which

timer  
to use:

CIA-A  
CIA-B  
DEVICE (timer.device)

#### SCHEDULER

Template: SCHEDULER/K  
Tooltype: SCHEDULER  
Default: STANDARD

Which

scheduler  
to use:

STANDARD  
MARKET  
REMEMBER  
QUEUES  
SUPER  
FEEDBACK

Specify NONE if you don't want to use any scheduler.

#### QUIT

Template: Q=QUIT/S  
Tooltype: QUIT

This will tell the server to quit and remove itself. For more information about quitting the server, see  
here

.

#### HELP

Template: HELP/S  
Tooltype: HELP

Display on-line help-file for the server.

#### VERSION

---

Template: VERSION/S  
 Tooltype: -

Display version number.

## 1.22 Commodity

COMMODITY

~~~~~

Executive is a commodity. You can disable/enable scheduling using a commodities exchange program. In the Exchange-utility, use the Active/Inactive button.

Executive doesn't have a GUI, so there's no popkey and the Show/Hide Interface buttons in Exchange have been disabled.

## 1.23 Quitting the server

QUITTING THE SERVER

~~~~~

You can quit the server by issuing this command:

Executive QUIT

or run the "Executive" program from Workbench and you'll be asked if you want quit Executive.

When the server quits, it will first signal  
     Acct  
     to write all unpurged  
 accounting records to disk.

You might get this message when you try to quit the server:

Unable to quit. Some program has patched a  
 function in Exec or Intuition library.

This means that you have started a program after the Executive server that has patched one or more of the same  
     functions  
     as the server.

The patches must be removed in reverse order, so you must check what program might have installed the patches. If this happens always when you try to quit Executive, then you should change the order of the programs you run in s:startup-sequence and s:user-startup. This requires some knowledge, so don't touch these files if you aren't sure what you are doing.

Another way to solve the problem is to use SetManager or PatchControl,

---

they are both available in Aminet:

```
setman10g.lha  util/boot  13K  Patches SetFunction() to new better one.
pchctl12.lha  util/wb     8K  PatchControl 1.2 - more control over pat
```

You might also get this message while trying to quit the server:

```
Unable to quit. Some clients are still running.
```

You must quit all clients before quitting the server. You might have left

```
Top
or
ALoad
running.
```

## 1.24 Preferences file

### PREFERENCES FILE

~~~~~

The preferences file is called "Executive.prefs". It's searched from the following directories in this order:

```
current directory
PROGDIR: (directory where the server binary is)
ENV:
S:
```

This is an ASCII file and contains one option on each line. Blank lines and lines starting with #-character are ignored. The option name and the value must be separated by spaces or tabs. Option names are case insensitive.

These options work like the CLI arguments and Workbench tooltypes, so consult their

```
documentation
```

.

```
CATCHMAX
CATCHMIN
DYNAMICMAX
DYNAMICMIN
NOFOCUS
SCHEDWB
TIMER
SCHEDULER
```

These options are used only in the preferences file:

```
TASK
CHILDTASKS
TASK+CHILDTASKS
IGNORE
```

## BASEDIR

## TASK

```

Template:  TASK <task name> <NOSCHEDULE> [priority]
                                                [BELOW]
                                                [ABOVE]
                                                <SCHEDULE> [nice-value]

```

Configures the server to watch for tasks with the given name. The name is case insensitive and \* can be used as a wildcard.

When a task matching the name is found, it's either NOT SCHEDULED or SCHEDULED. If you want the task NOT to be scheduled, you can optionally tell the server to keep the task's priority at the given value, or BELOW or ABOVE the dynamic range.

If the task is to be scheduled, you can give a default nice-value for it. Otherwise task will inherit its parent's nice-value.

## CHILDTASKS

```

Template:  CHILDTASKS <task name> <NOSCHEDULE> [priority]
                                                [BELOW]
                                                [ABOVE]
                                                <SCHEDULE> [nice-value]
                                                <RELATIVE>

```

Similar to TASKS option, but applies to all childtasks created by the named task. The RELATIVE option is explained [here](#)

.

## TASK+CHILDTASKS

```

Template:  TASK+CHILDTASKS <task name> <NOSCHEDULE> [priority]
                                                [BELOW]
                                                [ABOVE]
                                                <SCHEDULE> [nice-value]

```

Similar to TASKS and CHILDTASKS options, but applies to both the named task and all its childtasks.

## IGNORE

```

Template:  IGNORE <task name>

```

Configures the server to totally ignore the given task. Wildcards are supported and the name is case insensitive.

IMPORTANT! The <task name> is the actual task name, not the name of a CLI command. If task must be ignored, the server must do it at task creation time and then the CLI command name is not yet available.

This is needed only when a task is not removed with RemTask(). This is EXTREMELY rare and, illegal.

## BASEDIR

Template: BASEDIR <directory>

This option is created during installation. It's used for displaying on-line help so don't change it or delete it.

When you save the preferences file from

ExecutivePrefs

you'll lose the comments

you have in the preferences file. ExecutivePrefs add the default comments as seen below in this example preference file:

```
#### install directory (please don't delete this) ####
```

```
BASEDIR "SYS:Executive"
```

```
#### catch- and dynamic ranges ####
```

```
CATCHMAX 1
```

```
CATCHMIN -105
```

```
DYNAMICMAX -50
```

```
DYNAMICMIN -70
```

```
#### options ####
```

```
#NOFOCUS
```

```
#SCHEDWB
```

```
SCHEDULER FEEDBACK
```

```
TIMER DEVICE
```

```
#### tasks ####
```

```
TASK Top
```

```
SCHEDULE -15
```

```
CHILDTASKS ASwarm
```

```
NOSCHEDULE BELOW
```

```
TASK CygnusEd
```

```
SCHEDULE -20
```

## 1.25 RELATIVE

### RELATIVE

~~~~~

This is used together with the CHILDTASKS option. You can use it to schedule a group of tasks and retain their relative priorities. For example, you might have a parent task and three childtasks:

NAME	PRIORITY
-----	
ParentTask	0
Child1	1
Child2	-1
Child3	0

Executive would schedule these tasks, but it's can't guarantee that Child1 will always have higher priority than the ParentTask. But with the RELATIVE option this is possible. You might have this line in your

Executive.prefs file:

```
CHILDTASKS ParentTask RELATIVE
```

Executive will then schedule ParentTask, if its priority is in the catch range, and calculate new priority for it every second. But the childevents are not scheduled in the same way as before, instead their priority is calculated from ParentTask's priority. For example, if ParentTask gets priority -58, then childevent priorities will be:

```
Child1   -57   = ParentTask + 1
Child2   -59   = ParentTask - 1
Child3   -58   = ParentTask
```

If ParentTask is not scheduled, then all tasks will have their normal priorities.

If you don't want to schedule the ParentTask, it's ok to configure Executive to, for example, keep ParentTask's priority above scheduled tasks. But when RELATIVE option is used with ParentTask, you can't configure Executive to schedule / not schedule any of the childevents, that won't work.

There are also some other restrictions:

- \* ParentTask must not exit before its children.
- \* RELATIVE works for "first generation" childevents only. If Child1 creates a task Child1A, it will be scheduled in the usual way. Don't try to configure Executive to maintain relative priorities of Child1's childevents also, it's not allowed.
- \* Childevents look like they'd be totally idle, although they're not. CPU usage of the childevents is transferred to the parent, otherwise childevents could use as much CPU time as they like, and they'd always run with highest possible priority, blocking all other tasks.
- \* In the following situation Child1 and Child2 will both get priority ParentTask + 1, although Child2 should get priority ParentTask + 2.

NAME	PRIORITY
-----	
ParentTask	0
Child1	1
Child2	2

## 1.26 Server Internals

### SERVER INTERNALS

#### SCHEDULING

Executive schedules tasks by recalculating their priorities. This is

done once a second. Priorities are calculated only if the task has recently used CPU time.

## PID

PID means Process IDentifier. Every task has a unique PID, allocated by the server. If task is a CLI, the PID will be the CLI number. Otherwise a unique number from 200 upwards is given to the task.

PIDs are very useful when you need to change a nice-value or priority of some task. You don't have to type in the task name and sometimes there might be many tasks with the same name.

## NICE-VALUE

Each task has its own nice-value, usually 0. Negative nice-values (up to -20) give more CPU time to a task, i.e. it will be less nice to other tasks. Positive nice-values (upto +20) give less CPU time to a task. If you have two tasks that want to use all CPU time they get, and they have nice-values -20 and +20, the task with the +20 nice-value will still get some CPU time.

It's not possible to give any exact numbers of how much different nice-values have effect on task's priority. Each scheduler uses nice-values in somewhat different way and many other values affect the calculated priority.

## LOAD AVERAGES

Executive maintains three load averages, for the past 1, 5 and 15 minutes. A load average is the number of tasks that have been ready to run or running in the past 1, 5 or 15 minutes. If the load average is 1.00, you have had exactly one task running. If you have one task running and using 50% of CPU time, load average will be 0.50.

## WHAT TASKS SHOULD/SHOULDN'T BE SCHEDULED

There are very few tasks that shouldn't be scheduled. A good rule is not to schedule anything that has priority of 2 or higher. System tasks like input.device shouldn't be scheduled, and because their priority isn't in the catch range, by default they aren't scheduled.

Communication programs shouldn't be scheduled as they'll start to lose data due to limitations of the Amiga's serial port.

If you have a screenblinker that draws some fancy graphics on the screen while the screen is blanked, it will steal CPU time from other programs if it's scheduled, so it might be a good idea not to schedule it. Or, you can give it a nice-value +20, in which case it won't get as much CPU time as tasks with higher nice-values.

Some multi-function commodity programs like MultiCX and MCP have features (like SunMouse) that don't work properly if they don't get CPU time

---



almost immediately when they need it. Although the scheduler keeps these tasks running at a very high priority (because they don't use much CPU time) it's still possible that they become blocked by some higher priority task. So it's better to configure Executive so that these tasks aren't scheduled.

Please also read the  
Common problems  
section.

## FOCUS

Usually the task that has opened the currently active window is the one that receives user attention. Executive will give more CPU time to this task if you haven't turned the focus-feature off. The exact amount varies, depending on the scheduler you have chosen.

When you start the server, it will scan opened windows trying to figure out their owner, but it's not always possible, so it's best to start Executive before any windows have been opened. You can see the number of windows whose owner is known by using the  
Stat  
client.

## CHILD-PARENT

Executive maintains information about each task's parent and its children. It's not possible to figure out this relationship for tasks that are in the system when the server is started.

If a parent of some task terminates, its children will be transformed to the parent of the terminated task.

## PERFORMANCE

The server uses very little CPU time. Top and Ps show how much CPU time the server actually uses. Some of this time goes to serving clients. If you start 10 ALoad clients, you can see a tiny increase in server's CPU usage.

As a technical detail, Executive uses hash-tables to look up the internal information structure for each task, so there is no need to traverse long task lists.

So all in all, the cost of using Executive is insignificant. You'll easily spend more time changing priorities than Executive uses in a few weeks.

## REAL PRIORITIES

Executive keeps track of the real priorities of all tasks so it can restore them when you quit the server. The real priority is the one that is set with SetTaskPri() function, or the priority task had when

---

it was started.

#### STRANGE PRIORITIES

Exec.library lacks a function GetTaskPri(), which could be patched so the real priority of a task could be returned. Now it's possible that some task reads its scheduling priority and starts a childtask with this priority. The server will notice this and replace the priority with the correct value (parent task's real priority). This must be done, because if you will quit the server, childtask's priority would be incorrect.

Sometimes a childtask is started with parent priority - 1 (for example). The server can't correct this, but this is not a problem unless you quit the server. Then the task's priority will be incorrect, usually some negative value in dynamic range. But this is really very rare.

#### CONTEXT SWITCHES

Executive keeps track of the number of context switches each task makes. There are two types of context switches, voluntary and involuntary. Voluntary context switch happens when a task calls the Wait()-function and is put to sleep. Involuntary context switch happens when a task is forced to give the CPU to another task.

#### TIMERS

Executive needs to be signalled by a timer every second so it can recalculate priorities. Amiga has a device called "timer.device" which can be used to get accurate timing signals. There's just one problem: timer.device may cause hardware overflow errors in serial transfer, because Amiga's internal serial port is not buffered, it can only hold a total of one byte of incoming data.

There are two kinds of hardware timers in Amiga, CIA-A and CIA-B. Timer.device uses CIA-B. When a certain time has elapsed, the CIA timer causes an interrupt. CIA-B uses higher level interrupt than the serial interface, so it's served first. If the serial interface interrupt can't be served in time, a character is lost. CIA-A uses a lower level interrupt than CIA-B and the serial interface.

On Amigas with 68000 or 68010 Executive first tries to allocate the CIA-A timer. On Amigas with 68020+ Executive defaults to timer.device.

If Executive is configured to use one of the CIA-timers, and it can't allocate the timer, it will try the other one, and finally, if that can't be allocated either, it uses timer.device.

You can change the preferred timer using the TIMER option. If you don't have any problems with serial transfer, you may want to use timer.device, because then the CIA-timers will be left for other programs. For example, HippoPlayer and ShapeShifter can only use CIA-A timer.

To see what timer Executive is using, use the

```
Stat
-client.
```

OPEN WORKBENCH BUG FIX

See

```
DONFIXOPENWB option
for more information.
```

IS IT A HACK?

No, Executive is not a hack. Hack is something you do in five minutes. Executive has been constantly developed for three years.

HOW THE CPU USAGE FOR A TASK IS MEASURED?

The CPU usage calculation starts when a task is launched by the dispatcher and it's stopped just before the task is switched to another. The task time doesn't include the time that is spent in `exec.library` for saving the task's registers and finding a new task to run during a context switch. That's why the

```
Top
-client
```

rarely shows 100% CPU usage. If it's 99.7%, then the 0.3% of total available CPU time is spent in the task dispatcher doing "unproductive" work.

## 1.27 Clients

CLIENTS

~~~~~

Clients are utility-programs that are used together with the Executive server. Most of the clients require that the server is running. The Genies-directory contains

```
genies
, project icons that execute a client
```

with some tooltypes.

Here is a quick overview of included clients:

Acct

Accounting daemon

ALoad

Display load average or CPU usage

ALoad3D

Display load averages in 3D

---

|           |                                             |
|-----------|---------------------------------------------|
| Commander | General task manager                        |
| Ctp       | Extended ChangeTaskPri                      |
| Dashboard | The mother of all meters                    |
| Kill      | Extended Break                              |
| Lastcomm  | List last commands executed                 |
| Meter     | Display system information                  |
| Nice      | Run programs with lower scheduling priority |
| Ps        | Process status displayer                    |
| Pstree    | Display child-parent relationships          |
| Renice    | Renice a task                               |
| Sa        | Display accounting statistics               |
| Stat      | Display some misc. information              |
| Timer     | Time shell and Workbench programs           |
| Top       | Display information about the top CPU tasks |
| Uptime    | Display system uptime and load averages     |

All clients can be run from shell and Workbench. Most of the options can be given as CLI arguments or Workbench tooltypes.

You can use the HELP option with all clients and open this AmigaGuide documentation. For example:

```
Stat HELP
```

will show Stat's manual page. With clients that have a GUI or open a window, you can also press the Help-key or choose Project->Help from the menu.

---

Each client has a VERSION option, which can only be given when the client is started from a shell. It obviously displays the client's version number.

When you run a program from Workbench, the program opens a window for its output if necessary. The default size for this window is 640x200 pixels, but it can be changed with an environmental variable EXECUTIVEWINDOW. For example, if you have a PAL machine, you want a window which is 256 pixels high. You can set the environmental variable to the desired size with these commands:

```
setenv EXECUTIVEWINDOW "0/0/640/256"
copy ENV:EXECUTIVEWINDOW ENVARC:
```

The format of the environmental variable is left/top/width/height.

There are also programs which use Executive through the SysInfo.library

most of them are available in Aminet:

|             |                                 |                                    |
|-------------|---------------------------------|------------------------------------|
| lcdcpcu     | util/moni/lcdcpcu.lha           | Matthias Bethke < ←                |
|             | postmaster@sweetdreams.lahn.de> |                                    |
| MagicCX     | util/cdity/MagicCX12.lha        | Kai Iske <iske@informatik.uni- ←   |
|             | frankfurt.de>                   |                                    |
| rload       | Not released yet                | Bernhard Möllemann <zza@rz.uni- ←  |
|             | karlsruhe.de>                   |                                    |
| ScreenClock | util/time/ScreenClock.lha       | Thomas Igracki <T.Igracki@Bamp. ←  |
|             | berlinet.de>                    |                                    |
| SwazBlanker | util/blank/SwazBlanker240.lha   | David Swasbrook <swaz@pobox.com>   |
| TinyMeter   | util/wb/Tinymeter_V431.lha      | Tinic Urou <5uro@informatik.uni- ← |
|             | hamburg.de>                     |                                    |

These are the ones I'm aware of, please let me know if there are others.

## CLI ARGUMENTS

When a client is invoked with a question mark as an argument, it will output the argument template. For example, the command "Ps ?" will output this:

```
F=FORMAT, S=SORT/K, LIST/S, NH=NOHEADER/S, PH=PADHEX/S, PLAIN/S, DELAY/N/K,
IV=INVERSE/S, ACTIVE/S, IDLE/S, SCHED=SCHEDULED/S, RUN/S, READY/S, WAIT/S,
ALL/S, OWN/S, P=PID/K, U=USER/K, G=GROUP/K, N=NAME/K, R=PGRP/K, PRIGT/N/K,
PRILT/N/K, PRIEQ/N/K, NPROCS/N/K, EVERYTHING/S, BABY/S, TASK/S, PROCESS/S,
CLI/S, CPU/S, HELP/S, VERSION/S:
```

A CLI argument is divided in three parts:

```
NH=NOHEADER/S
^^ ^^^^^^^^ ^
| |         |
| |         |-- modifier
| |-- option name
|-- shortcut, can be used instead of the full name
```

The modifiers used in Executive are:

- /S - Switch. For example, "Ps SCHEDULED" will output only the scheduled tasks.
- /N - A decimal number.
- /K - Keyword. The option will not be understood unless the option name appears. For example if the template is "PGRP/K", then unless "PGRP=<pgpr>" or "PGRP <pgpr>" appears in the command line, PGRP will not be accepted.
- /A - Required. This keyword must be given a value during command-line processing, or an error is returned.
- /F - Rest of line. If this is specified, the entire rest of the line is taken as the parameter for the option, even if other option keywords appear in it.

If there are no modifiers after the option name, it's a string argument which can be specified without the option name. So you can invoke Ps with a format string in three ways:

```
Ps FORMAT=pid,cpu%,name
Ps F=pid,cpu%,name
Ps pid,cpu%,name
```

## 1.28 ScreenNotify

SCREENNOTIFY

~~~~~

To use Stefan Becker's ScreenNotify, copy the screennotify.library to LIBS:.

When ScreenNotify is in use, windows on Workbench or on any public screen will be closed before the screen closes, and reopened when the screen is again available. This lets you change, for example, Workbench screenmode or fonts while a program supporting ScreenNotify is running, without having to close the window first.

ScreenNotify is disabled while a requester is open, otherwise screens can't be opened or closed while the requester is being displayed.

If a screen is closed, but not opened again, and you had a window on that screen, you can reopen the window on default public screen by pressing the commodity hotkey, or using a commodities exchange program.

If you don't have ScreenNotify, you can FTP it from any Aminet site, for example ftp.wustl.edu. The file is:

```
util/libs/ScreenNotify10.lha
```

## 1.29 Genies

### GENIES

~~~~~

Genies are project icons that execute a client with some tooltypes. You can find them in the Genies-drawer. If you want to create your own Genie, just make a copy of one of the Genie-icons and change the default tool and tooltypes.

### USING NICE FROM WORKBENCH

You can run programs with a different nice-value by creating a Genie for the program you want to run. Use

Nice

as a default tool for the project

icon and put the name of the program you want to run to the COMMAND tooltype for example:

```
COMMAND=c:VirusChecker DH0:
```

### AVAILABLE GENIES

Here are the available Genies and their equivalent shell commands:

#### Scheduled tasks

Shell:

Ps

SCHEDULED

Display scheduled tasks.

#### Merge accounting

Shell:

Sa

MERGE

Summarize

accounting

and merge it with the (existing) summary database.

The current accounting file will be deleted. The summary is displayed on screen.

#### User accounting

Shell:

Sa

USER

Display per-user accounting summary. If you're not using MultiUser, this will just display all commands grouped together.

## 1.30 Schedulers

### SCHEDULER

~~~~~

Scheduler is a mathematical algorithm which calculates priorities for scheduled tasks. Scheduler uses different kinds of information when calculating priorities:

- \* system load average
- \* task's recent CPU usage
- \* task's context switches
- \* nice-value
- \* focus

Let me first say that it's not compulsory to use a scheduler, because even without a scheduler Executive is a quite powerful task manager. But you'll probably want to use one of the six different schedulers available. All schedulers on Executive are fair-share schedulers, they'll try to ensure that each task gets a chance to run, but favour tasks that don't use much CPU time, i.e. keep compute-bound tasks at low priority and transput-bound tasks at high priority. Schedulers recalculate task priorities once a second.

If the

focus option  
is on, the task which owns (i.e. has opened) the  
currently active window gets higher priority. Also, task's  
nice-value  
affects priority calculation.

If you aren't using a scheduler, then task priorities won't be changed by the server, but you can still use all other Executive features.

When a new task is added, it will be given some of its parent's scheduling information (e.g. recent CPU usage) and when a task exits, some of its scheduling information is transferred back to the parent. This is particularly useful when a task does most of its hard work in subtasks (AdPro for example uses subtasks for loading and saving images and when processing them).

To see how the system behaves under different schedulers use the

Ps  
and

Top  
clients and look how much CPU time different kinds of tasks get.

There are currently six different schedulers available:

STANDARD



When calculating task's priority, this scheduler considers task's recent CPU usage and system load average. There are no bells or whistles in this scheduler, task's priority is directly relative to its recent CPU usage.

This is the only scheduler available in the non-registered version of Executive.

## MARKET

This scheduler is based on "bidding" for CPU time. Task gets some money from the scheduler every second which it can use to "buy" CPU time. The task that has most money gets the first chance to run, i.e. the highest priority. When the system load average is higher, CPU time will cost more.

## REMEMBER

This is a "heavy-duty" scheduler, it can handle very highly loaded systems. It's similar to the STANDARD scheduler, but it will remember task's CPU usage for much longer time.

## QUEUES

This scheduler divides the dynamic priority range to eight priority levels and runs several tasks with the same priority, in the same Round-Robin queue. Tasks that use approximately the same amount of CPU time get same priority.

Exec will Round-Robin tasks that have same priority. Each task is run for a time-slice called quantum and then the next task gets chance to run. Exec normally uses a fixed quantum value 4, but this scheduler adjusts the quantum value for each task to maximize throughput of CPU intensive tasks and to make interactive tasks more responsive. Here are the quantum values given to tasks with different priorities:

PRIORITY	QUANTUM
above dynamic range	4
scheduled (inside dynamic range)	2-10
below dynamic range	12

For scheduled tasks the Quantum is the largest (10) when task's priority is close to the dynamicmax priority (by default -70).

This scheduler tries to eliminate "wobbling" priorities by not letting a task migrate upwards immediately. Task has to wait for one second before its priority can be raised. Focused task is not blocked this way.

Here you can see how tasks with approximately the same recent CPU-usage have the same priority:

PID	PRI	RPRI	NICE	TYPE	STATE	TIME	IDLE	CPU%	NAME
295	-50	1	0	proc	wait	0.801s	3m	0.00%	CygnusEd
287	-50	1	0	proc	wait	22.707s	6s	0.38%	CygnusEd
1	-62	0	0	bcli	wait	2.530s		4.94%	[find]
14	-62	0	0	bcli	wait	13.008s		7.09%	[find]
10	-75	0	0	bcli	ready	4.592s		13.91%	[lha]
15	-75	0	0	bcli	ready	22.401s		20.26%	[lha]

## SUPER

This is similar to the standard-scheduler, but it also uses

context switch counters

when calculating task's priority. Tasks

that use a lot of CPU time must usually be forced off the CPU involuntarily, so these tasks will be given lower priority than tasks which voluntarily give the CPU to another task. The result is that transput-bound tasks get more CPU time than compute-bound tasks.

## FEEDBACK

This is a real multi-level feedback queue scheduler, similar to what is used in most UNIX operating systems, for example in all BSD variants (NetBSD, FreeBSD, BSDI, etc.).

Like QUEUES, this scheduler also divides the dynamic priority range to multiple priority levels and runs several tasks with the same priority, in the same Round-Robin queue. Tasks that use approximately the same amount of CPU time get same priority. FEEDBACK uses nine queues.

See the QUEUES scheduler for explanation of Round-Robin and Exec quantum values (FEEDBACK uses same quantum values as QUEUES).

Executive recalculates task priorities every second, but this scheduler also constantly monitors running tasks' CPU usage, and if it seems that some task uses more CPU time than what is allowed on its current queue, drops task's priority immediately, so it won't block other tasks.

See the

server options

documentation for information on changing the

scheduler. Please note that it's not possible to change the scheduler on the fly, you must restart the server.

## 1.31 Accounting

ACCOUNTING

~~~~~

---

Most multiuser UNIX systems are set up to use accounting. The system administrator is able to see what programs have been executed, who has executed them and how much CPU time they have used. Accounting is very useful on single-user systems too. Have you ever wondered where all that CPU time goes to?

When accounting is enabled, server will create an accounting record every time a program started from shell or Workbench terminates. An accounting record consists of this information:

- Command name
- Time the command terminated
- Used CPU time
- Elapsed time
- User ID (if MultiUser has been installed)
- Group ID (if MultiUser has been installed)
- Miscellaneous flags

When enough accounting records have been created, or the oldest record has been in memory for a user specified time, the server signals the

```
Acct
-client and tells it to write the accounting records to disk.
```

Acct writes accounting records to a file called "acct". To see the last terminated commands, you use the

```
Lastcomm
-client. The most recently
terminated command is displayed first.
```

Because the "acct"-file may grow rather large in a short time, it's necessary to condense this information into a much shorter format and that is done by the

```
Sa
-client. Sa stands for summarize accounting.
```

Sa reads the "acct"-file and generates a summary that is written to files called "acct.procsun" and "acct.usersun" which contain statistics according to command name and user ID, respectively. If these files already exist then they are read and included in the summary. This information is included in the summary files:

- Command name (acct.procsun) or user ID (acct.usersun)
- Number of times the command has been executed (acct.procsun) or number of commands the user has executed (acct.usersun)
- Total used CPU time
- Total elapsed time

If MultiUser hasn't been installed, "acct.usersun" will contain only one user.

Normally Sa will just display the summary and not write it back to disk. To write the summary to disk, you must use the MERGE option with Sa.

Accounting records can't be created for internal commands of custom shells (CSH, SkSH, etc.).

---

## 1.32 Registration

### REGISTRATION

#### UNREGISTERED VERSION OF EXECUTIVE

These features require a registered version of Executive:

- \* All schedulers except STANDARD
- \* Accounting
- \* ALoad3D
- \* Meter

The absence of these features doesn't make Executive unusable. You can use Executive without these features but I hope you'd register after you have seen what this program is.

#### WHAT DO I GET WHEN I REGISTER?

You'll receive a  
keyfile  
which enables the disabled features. It  
will work with ALL FUTURE VERSIONS of Executive.

#### ORDERING

The easiest way to fill in a registration form is to use the "Register" program, which asks you all the necessary information and creates an ASCII registration form. A blank registration form is available in these formats:

Final Writer  
PostScript  
DVI (TeX)  
IFF  
ASCII

You'll find these files in the Register-directory.

If you don't have a printer, then create an ASCII registration form with the "Register" program and copy the necessary information to a piece of paper. The information I need is:

- your name
- address
- email address
- payment method, currency
- method of delivery

You can also email me the registration form and send the payment by

---

snail mail.

When you have filled in the registration form, please send it to me with your payment. My address is:

Petri Nordlund  
Vanhamaantie 4  
FIN-28800 PORI  
FINLAND

There's no need to pay extra for quick delivery, which seems to be common with some other shareware products. All registrations are processed within 24 hours.

Croatian users may register through Goran Paulin. The price is 90kn (Kuna). Fill in a registration form and send it with the payment to:

Goran Paulin, Rade Supica 1, HR-51000 Rijeka, CROATIA

For more information: Goran.Paulin@tvri.fido.hr, +385 (0)51 427 611.

#### REGISTRATION FEE

Registration fee is accepted in these currencies:

15 USD (US dollar)  
10 GBP (Pound sterling)  
20 CAD (Canadian dollar)  
20 AUD (Australian dollar)  
70 FIM (Finnish mark)  
20 DEM (Deutsche mark)  
100 NOK (Norwegian krone)  
100 SEK (Swedish krona)  
500 BEF (Belgian franc)  
70 FRF (French franc)  
25 NLG (Dutch guilder)  
100 DKK (Danish krone)

#### REGISTER TOGETHER WITH YOUR FRIENDS AND GET A DISCOUNT!

|                   |              |
|-------------------|--------------|
| 3-4 registrations | 20% discount |
| 5-7 "             | 30% "        |
| 8+                | 40% "        |

All registrations and the payment must be sent together. Note that I can't exchange coins at the local bank, so if you are sending cash, you have to round the payment to the nearest amount you can send without sending coins. Please don't hesitate to ask me if you are uncertain.

There's no need to send a registration form for each user, but the following information is needed:

- name

---

- address
- email address (if you have one)
- delivery:
  - by airmail on a 3.5" disk
  - keyfile emailed to you

The personalized keyfiles will be sent separately to each user.

#### PAYMENT METHODS

- Cash                      This is the preferred payment method and also the cheapest for you and me. Enclose the cash in two pieces of nonwhite paper to disguise it. Please DON'T SEND COINS, they cannot be exchanged at the local bank.
- Eurocheque                Write the cheque for 70 FIM, that is, 70 Finnish marks. You must write your card number on the back of the cheque, otherwise I can't cash it.
- Money transfer          Transfer 70 FIM to my account:
  - bank: Postipankki Ltd, Helsinki, Finland
  - account number 8000 29-26522193
  - pay via Eurogiro or SWIFT
  - SWIFT address: PSPBFIHH; Telex 121 698 pgiro fi
  - include a copy of the receipt with your registration form
- Postal Money             You can pay using postal money order, but you must Order                      mail or email me the registration form because it's possible that only your name is included with the PMO when I receive it, so I need your address also. Ask your local post office for more information.

This payment method is available for Finnish users only:

- Postiennakko            Postiennakkomaksu 20 mk lisätään hintaan (70+20=90 mk). Rekisteröintilomakkeen voit lähettää myös emailitse osoitteeseen <petrin@megabaud.fi>.

#### CHEQUES

As mentioned above, Eurocheques are accepted, but cashing other foreign cheques costs me 50 FIM (about 11 USD), so I prefer the other payment methods. But if you want to pay with a cheque, make it for this amount:

- 25 USD (US dollar)
  - 17 GBP (Pound sterling)
  - 34 CAD (Canadian dollar)
  - 34 AUD (Australian dollar)
  - 34 DEM (Deutsche mark)
  - 170 NOK (Norwegian krone)
  - 170 SEK (Swedish krona)
  - 850 BEF (Belgian franc)
  - 120 FRF (French franc)
-

42 NLG (Dutch guilder)  
170 DKK (Danish krone)

#### METHOD OF DELIVERY

If you have an Internet email address, I'll email you the keyfile on the same day I receive your registration. Otherwise I'll send you a 3.5" disk by post with the keyfile and the latest version of Executive.

The latest version of Executive can always be downloaded from Aminet. To find out what is the latest version, see the Executive WWW-page:

<http://www.megabaud.fi/~petrin/Executive.html>

Fidonet is very insecure, so I'll only transmit keyfiles through it if you send me your PGP-key.

### 1.33 Keyfile

#### KEYFILE

~~~~~

The keyfile contains your name, your address and a serial number. It's strictly forbidden to give this keyfile to anyone else. Several digital fingerprints are included in this file and redistributed copies are easy to identify.

When the Executive server is started, it will search for the keyfile from these directories:

current directory  
PROGDIR: (directory where the server binary is)  
ENV:  
S:

If you want to put the keyfile to encrypted partition or just somewhere else than in these directories, put the directory name to environmental variable "KEYPATH".

### 1.34 Glossary

#### GLOSSARY

~~~~~

##### blocked

A task that is ready but not running is blocked. This happens when a higher priority task is currently using the CPU.

---

### compute-bound

A task that spends most of its time just using the CPU without waiting for any external events is compute-bound. Opposite of transput-bound.

### catch range

Every task that has priority in this range is automatically scheduled by Executive. If task's priority is later changed into this range, Executive will start scheduling it.

### context switch

A context switch happens when the running task is switched to another. The context switch is involuntary if task didn't call Wait(), i.e. it had to be forced off the CPU.

### dynamic range

When Executive calculates priorities for scheduled tasks, the priorities will be in this range.

### Exec

Did you know that the full name of Exec is The Multitasking Executive?

### MultiUser

From the README-file included with MultiUser distribution:

MultiUser allows you to create a \*IX-like environment where several users live together in harmony, unable to delete each others files, unable to read those private love-letters of other users. And this even if several users are working on the system at the same time (on a terminal hooked up to the serial port)

People without a valid login ID and password won't be able to access files you have made private with MultiUser. If you make all files private (not readable for others), the only useful thing they could do, is boot from a floppy.

You can get MultiUser from any Aminet site, see the util/misc-directory.

### PID

Process IDentifier. Each task is given a unique identifier by the server. When manipulating tasks from shell, PID's are extremely useful.

### preemptive

Preemptive multitasking means that the operating system can force a task to give the CPU to another task.

### process

A process in Amiga is an extension of a task. The main difference is that only a process can use DOS to access files.

---



## Round-Robin

The way the OS shares CPU time between tasks with same priority. Each task gets to run for a specified time called "quantum", then the next task with same priority is run and the current task is moved to the end of the list.

## task

On other operating systems than AmigaOS tasks are called processes, but on Amiga process and task are different. Task is the basic execution thread.

## transput-bound

A task is transput-bound if it spends most of its time waiting for some event to occur. Opposite of compute-bound.

## 1.35 Developers

### DEVELOPERS

Executive has two interfaces for applications, SysInfo for statistics and ExecutiveAPI for controlling task scheduling. SysInfo is a shared library, ExecutiveAPI communicates with Executive via a public message port.

If you didn't install these packages when you installed Executive, you'll find them in the Developers.lzx file included in the Executive distribution archive.

If your program uses SysInfo or ExecutiveAPI, please let me know. And please ask me if you have problems with these. My email address is <petrin@megabaud.fi>.

## 1.36 Author

### AUTHOR

Executive has been a large project and I have sometimes lacked motivation to continue with its development, considering the uncertain future of Amiga, but things look much better now. The future of Amiga is still uncertain, but so it is also for PC's and other systems.

It's always nice to hear comments and suggestions about this program, as those have already given me a lot of motivation and driven the project further.

I intend to continue development and release new versions regularly, so

your thoughts and ideas are always welcome. Please tell me what you'd like to see implemented in Executive and what could be done better.

My Amiga is an old Amiga 2000, equipped with GVP G-Force 030/25 turbo, 340MB and 40MB harddrives and 10MB of memory. I also have a Citizen 120D printer and SupraFAX V.32bis modem.

I have also had Commodore 64, Amiga 500 and Atari ST1040 (which I quickly got rid of).

You can contact me in several ways:

Email

petrin@megabaud.fi  
petrin@vtoy.fi

WWW

Executive WWW page:

<http://www.megabaud.fi/~petrin/Executive.html>

My homepage:

<http://www.megabaud.fi/~petrin/>

Snail mail

Petri Nordlund  
Vanhamaantie 4  
FIN-28800 PORI  
FINLAND

Telephone

+358-2-6480 322 (EET)

PGP

For secure communication, please use PGP. Here's my  
PGP key

.

## 1.37 PGP KEY

PGP key

~~~~~

Below you will find my public key. Use a text editor to save it to disk and follow the instructions on PGP manual for using it to send encrypted email.

You can also obtain my PGP key by fingering my email address.

---

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
```

```
Version: 2.6ui
```

```
mQCNAjBz1LwAAEEALP5+1K/Mqbr7VDFrsL/W3SrplhRfp5GbLid6Gn38/Zfb1PJ
hb/yDAuJX0G9nZ/hMBS7nXPVuUPZsfliEeOuDW6U3MiqOrhau7mi6P05KqTW/xYY
fzTeZ+K6relfoR0+ScV4i0fYpMi/O4n+3uZGVGc4X/JSVQyHsTDgjUTe/EdpAAUR
tCNQZXRYaSBOb3JkbHVuZCA8cGV0cmcluQG1lZ2FiYXVklmZpPg==
=EBE0
```

```
-----END PGP PUBLIC KEY BLOCK-----
```

## 1.38 Background

### BACKGROUND

I started developing Executive after I had a debate about scheduling on AmigaOS and on OS/2. At the same time I also bought a bigger harddrive, so I could install NetBSD, the free UNIX operating system for Amiga. NetBSD has a "real" scheduler, and I wanted to have something similar in AmigaOS.

So I started the project. There weren't any similar programs available for Amiga, so I had to improvise. The first version used to calculate CPU usage by patching launch and switch vectors in every task structure. This was a bad idea and it wouldn't have worked with tasks that wanted to use these or the userdata field. So I changed the code so that CPU usage was calculated using a vertical blanking interrupt. I incremented a counter for each task if they were being run when the interrupt occurred. This worked and it was very efficient, but the accuracy was only 1/50 second, and in practice it was even less, because tasks that used CPU only between two vblanks didn't seem to use CPU at all. The current method for calculating CPU usage is explained elsewhere  
, so I won't repeat  
it here. It's now as accurate as is possible on AmigaOS.

## 1.39 Acct

### ACCT

Accounting daemon.

```
OPTIONS
DESCRIPTION
```

```
Acct writes
    accounting
```

records to disk. Server creates an accounting record every time a program started from shell or Workbench terminates.

To reduce disk access Acct purges records to disk only after certain amount of them has been created or when the oldest record has been in memory for a user specified time. Acct doesn't use any CPU time when it's waiting for records.

The easiest way to start Acct is to copy it to your WBStartup-drawer. The Acct-client's icon should include a DONOTWAIT-tooltype, which tells Workbench not to wait for Acct to exit before proceeding.

It's not necessary to put Acct to WBStartup-drawer, you can start it anytime you want to, from Workbench or from shell. If you start it from shell, you have to run it:

```
run <NIL: >NIL: Acct
```

The default accounting file is "s:acct", but you may change this. Use the ACCTFILE option to do this or set the environmental variable ACCTFILE to contain the file name.

If the disk where accounting records are being written becomes full, accounting records might be lost. Accounting won't be disabled so you can make some space to the disk and Acct will then continue normally. On heavily loaded systems 100-200KB of accounting records can be created each day, but a normal amount is 5-20KB.

To turn accounting off, send a break-signal to the accounting daemon or use the command:

```
Acct OFF
```

If you're using Workbench, start the Acct-client again and it will ask you if you want to turn accounting off.

If Acct notices an unprintable character in the command name, it will be changed to "?" to prevent cluttered display.

Acct works by patching the dos.library RunCommand() function. If you run another program that patches this function, it's not possible to quit Acct before the patch is removed. This is not very likely so you don't have to worry about it. A more common problem when quitting Acct is that some commands are still being executed and they have called the RunCommand() function, so it's not possible to safely remove the patch. If the patch can't be removed, Acct will notify you about this. Accounting will be disabled and Acct stays in memory waiting until the patch can be removed. You don't have to worry about this, you can safely start Acct again if necessary.

The accounting system tries to distinguish commands run in a shell and programs started from Workbench, other tasks are not accounted. If you wish that all tasks should be accounted, use the ALL option. There are also other ways to start programs, for example with WBRUN and WBSTART. All versions of WBSTART are supported, and programs started using it are properly accounted. WBRUN is not currently

---

supported, as it uses a totally different method for starting programs.

Acct supports PatchControl.

#### EXAMPLES

Start accounting and output all records to screen instead of writing them to disk. Press CTRL-C to stop.

Acct STDOUT

## 1.40 Acct options

### ACCT OPTIONS

#### ACCTFILE

Template: F=ACCTFILE  
Tooltype: FILE  
Default: s:acct

The accounting file name.

#### OFF

Template: OFF/S  
Tooltype: OFF

Turn accounting off.

#### SECONDS

Template: S=SECONDS/N/K  
Tooltype: SECONDS  
Default: 30

Number of seconds to wait before writing records to disk. You can set this to 0, but then the disk must be accessed every time a command terminates and this may not be a good idea, unless you have a disk cache.

#### RECORDS

Template: R=RECORDS/N/K  
Tooltype: RECORDS  
Default: 10

The maximum number of accounting records in memory at once. When this limit is reached, records will be written to disk.

#### STDOUT

Template: STDOUT/S  
Tooltype: STDOUT

Instead of writing records to disk, display them on screen. This will also turn the COOKED option on. See

Lastcomm  
for the output format.

#### COOKED

Template: C=COOKED/S  
Tooltype: COOKED

Write accounting records in human-readable form, not in binary. If you write records to disk in this format, you can't use

Lastcomm  
or  
Sa  
to manipulate them. See  
Lastcomm  
for the output format.

#### ALL

Template: ALL/S  
Tooltype: ALL

Create accounting records for ALL tasks, not just programs started from shell or Workbench.

#### PURGE

Template: PURGE/S  
Tooltype: PURGE

Force Acct to write all records currently in memory to disk. Do this before resetting your system.

#### HELP

Template: HELP/S  
Tooltype: HELP

Display this document.

#### VERSION

Template: VERSION/S  
Tooltype: -

Display version number.

---

## 1.41 ALoad

### ALOAD

~~~~~

ALoad displays a periodically updating histogram of system load average or current CPU usage.

### OPTIONS DESCRIPTION

Two versions of ALoad are provided, a GadTools-version for all Amigas and one which supports MUI.

ALoad opens a small window and displays a periodically updating histogram of the system load average over last minute, or current CPU usage.

When load average reaches 1.0, the window is divided in two with a line through the middle. This line represents load average 1.0, the top part of the window represents load averages 1.0-1.999... When load average reaches 2.0, the window will be divided in three sections. This can be controlled with the SCALESTEP option.

The graph is scrolled to left when it reaches the right border. You can resize the window without losing the old histogram, which will be redrawn to the new window.

The GAUGE option displays a small car speedometer like meter of current CPU usage. The meter is updated four times a second. A small damping factor is used to slow down the meter's needle. Because the MUI V3 Levermeter-class is very small, no label is displayed for the window. There's no size gadget either because the class is not scalable. The Scroll, Update, Scale and Type menus are disabled. You'll probably want to set the window border width to 0 using MUI preferences.

ALoad can optionally display a clock in the window's titlebar.

There are several menu items available:

Project->New will redraw the histogram.

Settings->Update sets the update rate in seconds.

Settings->Scroll sets the amount that the histogram is scrolled when it reaches the right edge. 1/2 means that the histogram will be scrolled half the width of the current window.

Settings->Scale sets the MINIMUM scale of the histogram, i.e. the minimum number of divisions in the scale.

Settings->Type is used to choose what value is displayed, system load average or current CPU usage.

---

Settings->Clock toggles the titlebar clock on/off.

Settings->Stay front makes the window stay in front of all other windows.

Settings->MUI Preferences... opens then MUI preferences editor.

The Project->Save config menu item will save current settings to the program's icon. This option is available only when the program is started from Workbench.

The non-MUI version of ALoad has invisible drag, close, depth and size gadgets if the system gadgets have been disabled with NOTITLEBAR or NOSIZEGADGET options. The centre of the window works as a drag-bar.

ALoad is a commodity and supports (GadTools-version only)

```
screennotify.library
```

```
.
```

## EXAMPLES

Load average:

```
ALoad
```

Current CPU usage:

```
ALoad CURRENT
```

Scroll 10 pixels, update every 30 seconds, don't display a size gadget:

```
ALoad SCROLL=10 UPDATE=30 NOSIZEGADGET
```

## 1.42 ALoad options

```
ALOAD OPTIONS
```

### SCROLL

```
Template: SCR=SCROLL/N/K
```

```
Tooltype: SCROLL
```

```
Default: 1
```

The number of pixels to shift the graph to the left when the graph reaches the right edge of the window. If you specify value 0 here, it means that the histogram will be scrolled half the width of the current window.

### UPDATE



Template: UD=UPDATE/N/K  
Tooltype: UPDATE  
Default: 1

The update interval in seconds.

#### SCALE

Template: SCA=SCALE/N/K  
Tooltype: SCALE  
Default: 1

The minimum number of tick marks in the histogram, where one division represents one load average point. If the load goes above this number, ALoad will create more divisions, but it will never use fewer than this number.

#### SCALESTEP

Template: SCS=SCALESTEP/N/K  
Tooltype: SCALESTEP  
Default: 1

Draw only every SCALESTEP'th scaleline. If SCALESTEP is 3, every third scaleline is drawn and the display is scaled so that load averages 0 - 2.999... are displayed.

#### CURRENT

Template: C=CURRENT/S  
Tooltype: CURRENT  
Default: display system load average

Display current CPU usage instead of system load average.

#### BGCOLOR

Template: BGC=BGCOLOR/N/K  
Tooltype: BGOLOR

Background color. (0-255)

#### COLOR

Template: CO=COLOR/N/K  
Tooltype: COLOR

Graph color. (0-255)

#### SCALECOLOR

Template: SCC=SCALECOLOR/N/K  
Tooltype: SCALECOLOR

Scale line color. (0-255)

#### CLOCK

---

Template: CLOCK/S  
Tooltype: CLOCK

Display a clock in window titlebar.

#### LABEL

Template: LB=LABEL/K  
Tooltype: LABEL

The text in the window's titlebar. You can put here whatever you like.

#### BACKDROP

Template: BACK=BACKDROP/S  
Tooltype: BACKDROP

Backdrop window, the window is behind all other windows and can't be depth arranged.

#### NOTITLEBAR

Template: NTB=NOTITLEBAR/S  
Tooltype: NOTITLEBAR

Don't display a titlebar for the window. This option is not available in the MUI version of ALoad.

#### NOSIZEGADGET

Template: NSG=NOSIZEGADGET/S  
Tooltype: NOSIZEGADGET

Don't display a size gadget for the window.

#### CX\_PRIORITY

Template: CX\_PRIORITY/N/K  
Tooltype: CX\_PRIORITY  
Default: 0

Commodity priority (from -128 to 127). This option is not implemented in the MUI-version of ALoad.

#### CX\_POPUP

Template: CX\_POPUP/K  
Tooltype: CX\_POPUP  
Default: yes

This can be "yes" or "no". If it's "no", then ALoad will not open its window when it is started. This option is not implemented in the MUI-version of ALoad.

#### CX\_POPKEY

---

Template: CX\_POPKEY/K  
Tooltype: CX\_POPKEY  
Default: no hotkey

The hotkey which will open ALoad's window closed with the Project->Hide command. For example, "ctrl alt a". This option is not implemented in the MUI-version of ALoad.

#### NOSCREENNOTIFY

Template: NOSCREENNOTIFY/S  
Tooltype: NOSCREENNOTIFY

Don't use  
    screennotify.library  
    . This option is not implemented  
in the MUI-version of ALoad.

#### STAYFRONT

Template: SF=STAYFRONT/S  
Tooltype: STAYFRONT

Keep the window in front of all other windows.

#### LEFT

Template: LEFT/N/K  
Tooltype: LEFT

Offset of the left edge of the window relative to screen.  
This option is not implemented in the MUI-version of ALoad.

#### TOP

Template: TOP/N/K  
Tooltype: TOP

Offset of the top edge of the window relative to screen.  
This option is not implemented in the MUI-version of ALoad.

#### WIDTH

Template: WIDTH/N/K  
Tooltype: WIDTH  
Default: Calculated from display width

Width of the ALoad window. This option is not implemented in the MUI-version of ALoad.

#### HEIGHT

Template: HEIGHT/N/K  
Tooltype: HEIGHT  
Default: Calculated from display height

Height of the ALoad window. This option is not implemented in

---

the MUI-version of ALoad.

#### PUBSCREEN

Template: PS=PUBSCREEN/K  
 Tooltype: PUBSCREEN  
 Default: default public screen

Name of the public screen where the window is to be opened.  
 This option is not implemented in the MUI-version of ALoad.

#### HELP

Template: HELP/S  
 Tooltype: HELP

Display this document.

#### VERSION

Template: VERSION/S  
 Tooltype: -

Display version number.

## 1.43 ALoad3D

### ALOAD3D

~~~~~

ALoad3D displays a periodically updating 3D bar-chart of system load averages.

#### OPTIONS DESCRIPTION

ALoad3D opens a small window and displays a periodically updating three dimensional bar-chart of the system load averages over last minute, last 5 minutes and last 15 minutes.

The maximum displayed load average is 5.0. If a load average is higher than this, then a small pyramid is generated on top of the bar. Server recalculates load averages once a second.

The thickest bar is the 1 minute load average and the thinnest bar represents the 15 minute load average.

There are several menu items available:

Settings->Delay between updates sets the delay in 1/50 seconds between display updates.

Settings->Speed sets the rotation speed (degrees).

Settings->Grow selects old style bars.

Settings->Motion->None stops the rotation.

Settings->Motion->Rotate rotates the bar-chart.

Settings->Motion->Wiggle makes the bar-chart rotate from left to right and then from right to left.

Settings->Stay front makes the window stay in front of all other windows.

The Project->Save config menu item will save current settings to the program's icon. This option is available only when the program is started from Workbench.

You can rotate the bar-chart using left and right cursor keys.

The window has invisible drag, close, depth and size gadgets if the system gadgets have been disabled with NOTITLEBAR or NOSIZEGADGET options. The centre of the window works as a drag-bar.

ALoad3d is a commodity and supports  
screennotify.library  
.

ALoad3D requires a registered version of Executive.

## EXAMPLES

ALoad3D

## 1.44 ALoad3D options

### ALOAD3D OPTIONS

#### DELAY

Template: D=DELAY/N/K  
Tooltype: DELAY  
Default: 3

Delay between display updates in 1/50 seconds. Maximum delay is 50 (one second). If the delay is 0, display will be constantly updated.

#### SPEED

Template: S=SPEED/N/K  
Tooltype: SPEED  
Default: 3

The rotation speed, maximum is 10.

#### GROW

Template: GROW/S  
Tooltype: GROW

Use old style bars.

#### NOMOTION

Template: NM=NOMOTION/S  
Tooltype: NOMOTION

Don't rotate the bar-chart.

#### WIGGLE

Template: WG=WIGGLE/S  
Tooltype: WIGGLE

Makes the bar-chart rotate from left to right and then from right to left.

#### ANGLE

Template: A=ANGLE/N/K  
Tooltype: ANGLE  
Default: 0

Default rotation angle.

#### LABEL

Template: LB=LABEL/K  
Tooltype: LABEL

The text in the window's titlebar. You can put here whatever you like.

#### BACKDROP

Template: BACK=BACKDROP/S  
Tooltype: BACKDROP

Backdrop window, the window is behind all other windows and can't be depth arranged.

#### NOTITLEBAR

Template: NTB=NOTITLEBAR/S  
Tooltype: NOTITLEBAR

Don't display a titlebar for the window.

#### NOSIZEGADGET

Template: NSG=NOSIZEGADGET/S

---

Tooltype: NOSIZEGADGET

Don't display a size gadget for the window.

#### CX\_PRIORITY

Template: CX\_PRIORITY/N/K

Tooltype: CX\_PRIORITY

Default: 0

Commodity priority (from -128 to 127).

#### CX\_POPUP

Template: CX\_POPUP/K

Tooltype: CX\_POPUP

Default: yes

This can be "yes" or "no". If it's "no", then ALoad3D will not open its window when it is started.

#### CX\_POPKEY

Template: CX\_POPKEY/K

Tooltype: CX\_POPKEY

Default: no hotkey

The hotkey which will open ALoad3D's window closed with the Project->Hide command. For example, "ctrl alt 3".

#### NOSCREENNOTIFY

Template: NOSCREENNOTIFY/S

Tooltype: NOSCREENNOTIFY

Don't use

screennotify.library

.

#### STAYFRONT

Template: SF=STAYFRONT/S

Tooltype: STAYFRONT

Keep the window in front of all other windows.

#### LEFT

Template: LEFT/N/K

Tooltype: LEFT

Offset of the left edge of the window relative to screen.

#### TOP

Template: TOP/N/K

Tooltype: TOP

Offset of the top edge of the window relative to screen.

#### WIDTH

Template: WIDTH/N/K  
Tooltype: WIDTH  
Default: Calculated from display width

Width of the ALoad3D window.

#### HEIGHT

Template: HEIGHT/N/K  
Tooltype: HEIGHT  
Default: Calculated from display height

Height of the ALoad3D window.

#### PUBSCREEN

Template: PS=PUBSCREEN/K  
Tooltype: PUBSCREEN  
Default: default public screen

Name of the public screen where the window is to be opened.

#### HELP

Template: HELP/S  
Tooltype: HELP

Display this document.

#### VERSION

Template: VERSION/S  
Tooltype: -

Display version number.

## 1.45 Commander

### COMMANDER

~~~~~

A general task-manager with GUI.

### OPTIONS DESCRIPTION

Two versions of Commander are provided, a GadTools-version for all

---



Amigab and one which supports MUI.

Commander opens a window showing all tasks currently in system. The display looks like this:

| NAME           | PID | TYPE | PRI | RPRI | NICE |
|----------------|-----|------|-----|------|------|
| [CyberCron]    | 4   | bcli | -70 | 0    | 0    |
| console.device | 208 | task | 5   | 5    | 0    |
| DH0            | 247 | proc | -63 | 1    | -20  |
| ramlib         | 209 | proc | -70 | 0    | 0    |
| ...            |     |      |     |      |      |

After the task name and its PID, the TYPE field shows if the task is a task, process, cli or background cli. PRI is the task's priority, RPRI is its real priority, the one that will be restored when the server quits. NICE is, obviously, task's nice-value.

If the task name is in square brackets, it's a shell command.

#### BREAK

The break-button lets you send a break-signal to the selected task. You can choose from four different signals, usually you'll use the C signal, which is used to break a task. D-signal is used with shell scripts and E-signal is used with commodity-programs. F-signal is currently undefined. Note that Commander only sends the break-signal to a task, it can't guarantee that the task will actually exit.

#### NICE

This lets you modify tasks nice-value. Higher values make the task "nicer", i.e. it will be given less CPU time.

#### SIGNAL

Each task can receive 32 different signals. If you don't know anything about signals, press the cancel-button and don't try to signal a task, it will probably just crash your system.

All the 32 signals are shown in the window. You can choose any number of signals you want to send. If a signal is shown in white, it means that the task is currently waiting for this signal. If the signal has been ghosted, it's not allocated and sending it has no effect.

#### REMOVE

Will remove the selected task from exec.library task-list by calling RemTask(). Task's resources won't be deallocated and a crash is very likely. Don't use this if you don't know exactly what you're doing.

#### PRIORITY

Set the priority of a task. If you change task's priority so that it's in the catch range, scheduler notices this and starts scheduling the task.

#### UPDATE

Update the task list. The task list is not updated automatically, so it may include tasks that have long since terminated.

The MUI version of Commander also displays CPU usage of the selected task in a graph on top of the window. The default is to show recent CPU usage, but choose the Settings->MeterType->Current CPU usage menu item to show last second CPU usage.

The Project->Save config menu item will save current settings to the program's icon. This option is available only when the program is started from Workbench.

Settings->MUI Preferences... opens then MUI preferences editor.

## EXAMPLES

Commander

## 1.46 Commander options

### COMMANDER OPTIONS

~~~~~

#### CURRENT

Template: C=CURRENT/S  
Tooltype: CURRENT

Instead of showing recent CPU usage, show current, last second CPU usage. This option is available only in the MUI-version of Commander.

#### ZIP

Template: ZIP/S  
Tooltype: ZIP

Open Commander's window so that only the titlebar is displayed. This option is available only in the GadTools-version of Commander.

#### LEFT

Template: LEFT/N/K  
Tooltype: LEFT

Offset of the left edge of the window relative to screen. This option is not implemented in the MUI-version of Commander.

#### TOP

Template: TOP/N/K  
Tooltype: TOP

---

Offset of top edge of the window relative to the screen.  
This option is not implemented in the MUI-version of Commander.

#### HELP

Template: HELP/S  
Tooltype: HELP

Display this document.

#### VERSION

Template: VERSION/S  
Tooltype: -

Display version number.

## 1.47 Ctp

### CTP

~~~~~

Change task's priority.

#### OPTIONS DESCRIPTION

Ctp is similar to the standard Amiga utility ChangeTaskPri, but it has a lot more options.

Where ChangeTaskPri can only change the priority of one task at a time, Ctp can change the priority of multiple tasks with one command. For example:

```
Ctp PRI=-1 PID=10,20 NAME=DH\* USER=root GROUP=staff PGRP=273
```

The following tasks will be affected by this command:  
(NOTE: Depending on the shell you use, the \ may not be needed in front of the asterisk.)

- tasks with PID 10 and 20
- tasks with name starting with "DH" (case insensitive)
- all tasks owned by the root user. (MultiUser only)
- all tasks owned by the staff group. (MultiUser only)
- all tasks that belong to process group 273.

All these tasks get the specified priority, -1.

If you don't specify any task, priority of the current task (usually a shell process) will be changed.

There's no need to use the PRI- and PID-keywords, Ctp works without them too:

```
Ctp -10 278
```

This will change the priority of the task with PID 278 to -10.

You can replace the AmigaOS command `c:ChangeTaskPri` with `Ctp`, they are compatible and `Ctp` doesn't need the server to run. If the server isn't running you can use only PIDs that correspond to CLI numbers

.

When `MultiUser` is installed, only the super-user may raise task's priority. If you don't want to require root privileges for `Ctp`, set the U-bit using `MProtect`.

#### EXAMPLES

Change the priority of the current task to 0.

```
Ctp 0
```

Change the priority of "RexxMaster" to 5.

```
Ctp 5 NAME=RexxMaster
```

## 1.48 Ctp options

#### CTP OPTIONS

##### PRIORITY

```
Template: PRI=PRIORITY/N/A
Tooltype: PRIORITY
```

New priority for the specified tasks.

##### PID

```
Template: PID=PROCESS
Tooltype: PID
Default: Current task
```

PIDs of the tasks whose priority is to be changed. You can list multiple PIDs here, separate them with a comma, for example "PID=10,248,332".

##### NAME

```
Template: N=NAME/K
Tooltype: NAME
```

Names of the tasks whose priority is to be changed. You can list multiple names here, separate them with a comma, for example "NAME=DH0,DH1,DH2,

\*.device". You can use \* as a wildcard.

#### USER

Template: U=USER/K  
Tooltype: USER

Names of the users whose all tasks will get the new priority. You can list multiple user names here, separate them with a comma, for example "USER=Frodo,Gandalf,Aragorn". You can also use UIDs. This option isn't available if MultiUser hasn't been installed.

#### GROUP

Template: G=GROUP/K  
Tooltype: GROUP

Names of the groups whose all tasks will get the new priority. You can list multiple group names here, separate them with a comma, for example "NAME=staff,students". You can also use GIDs. This option isn't available if MultiUser hasn't been installed.

#### PGRP

Template: R=PGRP/K  
Tooltype: PGRP

PGRPs of the process groups whose priority is to be changed. All tasks in the process group will be affected. You can list multiple PGRPs here, separate them with a comma, for example "PGRP=254,298,578".

#### HELP

Template: HELP/S  
Tooltype: HELP

Display this document.

#### VERSION

Template: VERSION/S  
Tooltype: -

Display version number.

## 1.49 Dashboard

### DASHBOARD

~~~~~

The mother of all meters.

### OPTIONS

---

## DESCRIPTION

The best way to learn to use Dashboard and all the configuration options it offers is to play with it. You can't break it, hopefully. :)

Introduction

Getting started

Common elements

Global settings

Windows

Meters

ScreenNotify

Tips & tricks

EXAMPLES

Dashboard

## 1.50 Dashboard - Introduction

### DASHBOARD - INTRODUCTION

~~~~~

Dashboard lets you open windows on public screens and create any number of meters on each window.

Meters display various values like CPU usage, free memory, load average, etc. The size and position of a meter can be adjusted with mouse while the program is running.

Windows and meters can be individually saved or just closed and then easily reopened later.

Dashboard is extremely configurable. All configuration options are in menus.

Some example preference files are included in the DashboardPrefs-directory:

|                |                                |
|----------------|--------------------------------|
| ALoad          | ALoad clone                    |
| Clock_Calendar | Display a clock and a calendar |
| CPU_Memory     | CPU usage and free memory      |
| SANA_II-Status | SANA-II device status          |
| VMM            | Virtual memory (VMM) status    |
| WorldTime      | Time around the world          |

It's intended that the user creates own preference files, as they

depended very much of the available screenmodes, colors and fonts. That's why Dashboard is so configurable. These files are just meant to get you started.

These should be used on a screen with pixel aspect 1:1, e.g. 640x512. Helvetica and Times fonts are used, so they should be in your FONTS: directory.

CPU\_Memory requires OS3.x or later, because it uses datatypes to load images. A 16+ color screen is required for the images to look normal.

After loading "WorldTime", select from menu Global->Local timezone->Set to set your local timezone.

The SANA-II status meter has no device set by default. To set the name, select one of the meters and choose from menu SANA-II status->Device... and give the correct device name. Remember that device names are case sensitive. The SANA-II devices are usually in "DEVS:networks"-directory, so you'd enter for example "DEVS:networks/ppp.device". You'll have to change the name for one meter only. The On-line meter displays how long you have been on-line. It defaults to "serial.device", which has to be opened in SHARED mode by you communication program or AmiTCP.

## 1.51 Dashboard - Getting started

### DASHBOARD - GETTING STARTED

When Dashboard starts, it loads the default preferences file, "DashboardPrefs/Default". The DashboardPrefs-directory is in the directory where the program binary is, or in a user specified directory (PREFSDIR option). A preferences file contains global settings and one or more windows.

The Project->Open and Project->Insert commands are used to load preference files from disk. You can either replace the current windows and settings or just add the windows from the preferences file.

When you save the current configuration with Project->Save as... command, an icon will be created if the Project->Options->Save icons option is on.

We'll look at windows and meters later, but just to get you started, here's how you can create a new window and put some meters to it:

Open a new window by selecting: Window->New window.

Meters have been divided into four categories: Global, Task, Storage and Misc. Select Meters->Global->CPU usage.

The mouse pointer should now change to a crosshair. Click anywhere on the window and by holding down the left mouse button, drag a small box on the window. When you let go of the mouse button, a CPU usage meter is created.

To move this meter to another position, just drag it there with mouse. You can adjust its size from any of the corners or edges.

Create another meter, select: Meters->Misc->Calendar.

Now try resizing the window.

You can select meters with mouse, or unselect them by clicking somewhere on the window. You can also "lasso" them.

Select the CPU usage meter and then select from menu: Graph->Type->Round.

Keep the meter selected and choose: CPU Usage->Update->1/4. The meter will now be updated four times per second.

To make the meter look a bit better, select CPU Usage->Fill->Dither and CPU Usage->Frame->Title.

Try Window->Fill->Chess and then Window->Fill->Size..., enter 10 and press Ok.

Ok, you have only scraped the surface, but you'll learn more later.

## 1.52 Dashboard - Common elements

### DASHBOARD - COMMON ELEMENTS

Colors, fonts, frames and fills are common elements used in Dashboard windows and meters.

Dashboard has three kinds of colors and fonts:

```

Global default colors and fonts
  \
  The same colors and fonts for each window
    \
    Colors and fonts used by meters
  
```

By default, colors and fonts used by meters point to window colors and fonts, which in turn point to global default colors and fonts.

If you change a global default color or font, the change will affect all colors and fonts that point to it. For example, there's a color called "Light edge color", which is used for highlighted edges. There's also a "Light edge color" for each window, which by default points to the global default "Light edge color", so changing the global color will also change the window "Light edge color". This same color is also used when drawing frames for meters. You can of course individually change the "Light edge color" for each window and meter frame.



## COLORS

Colors can be adjusted from menus, three choices will be available for each color:

```
Color...
Pen...
Colorwheel...
```

Color lets you choose a fixed color from the screen palette. Pen allows you to choose one of the standard Intuition pens. Colorwheel requires OS3+, it lets you choose a color which Dashboard tries to allocate from the system. If the exact color is not available, the closest color is used.

For some colors there's also a Default-item in the menu. This will point to this color's default color, for meter color this points to one of the window colors and for window colors, this points to a global default color.

Some colors can be transparent. If this is possible, there will be a Transparent-item in the color menu.

## FONTS

The global default font is set to screen font, the default font for each window is the font used by the screen where the window is.

Fonts can also be chosen from menus, for all fonts, there are at least two menu items, Select... and Screen font. For some fonts there is also a Default-item in the menu. Select it, and this font's default font will be used.

## FRAMES

Dashboard uses the same frame drawing routine for window edges, meter edges, and some meter types use it, e.g. to draw a frame for a bar gauge. No matter where a frame is used, there is always a similar menu for the frame:

```
Frame >>  None
          Simple
          V Bevelled
          Button
          Double
          Ridge
          Title
          Intuition window borders
          ~~~~~
          Size...
          Extra space...
            Inverted
            Filled
          ~~~~~
          V Top
```

V Bottom  
 V Left  
 V Right

The first items indicate the frame type:

|          |                                                                              |
|----------|------------------------------------------------------------------------------|
| None     | no frame                                                                     |
| Simple   | a simple one-color rectangle                                                 |
| Bevelled | a bevelled rectangle                                                         |
| Button   | like Bevelled, but the pixels at bottom left and top right corners are drawn |
| Double   | two bevelled frames                                                          |
| Ridge    | this kind of frame is used in string gadgets                                 |
| Title    | a Ridge-like frame, but meter name is displayed at the top                   |

Intuition window border is only available for the window frame. This is the standard window border.

Size... is used to set the size of a double frame, the amount of space between the two rectangles. If double frame is used for window frame, this space will be filled when the window is selected.

Extra space... is used to add some extra space between the frame and, for example, a meter.

Inverted exchanges the two edge colors.

Filled always fills the area specified by Size... and Extra space....

The Top, Bottom, Left and Right items are used to turn on/off the frame edges. You can, for example, put two meters one below the other, with no space in between, and turn off the top and bottom frame edges so that it looks like both meters are enclosed by a single frame.

There's also a Frame colors-menu, you'll find the two edge colors and the fill color from there.

## FILLS

Fills are used to fill the window or meter background. A similar menu is used for all fills:

```

Fill >>  None
         Color
         Raster
         Chess
V Dither
         Image...
~~~~~
Size...
         Remap
         Scale
~~~~~

```

```

==== COLOR 1 ====
  Color...
  V Pen...
  Colorwheel...
==== COLOR 2 ====
  Color...
  V Pen...
  Colorwheel...

```

The first items are the fill type:

```

None          no fill
Color         simple one-color fill
Raster        2x2 raster using the two colors
Chess         chessboard pattern
Dither        vertically dithered (ordered dithering)
Image...     Custom image

```

Image... is only available on OS3+. When you select this type, a file requester will open where you can choose an image to be used. The image is loaded using datatypes. Image can be used with a window or meter fill, not with others.

Size... sets the chessboard pattern size.

Remap - the image can be remapped to the screen palette.

Scale - the image can be scaled to the fill size, if enough memory is available, otherwise the image will be tiled.

## 1.53 Dashboard - Global settings

DASHBOARD - GLOBAL SETTINGS

~~~~~

From the Global menu you adjust global default colors and fonts, and the local timezone.

```

Background color  Used for various backgrounds
Light edge color  Frame edge - light
Dark edge color   Frame edge - dark
Label color       Used for meter's label
Value color       Used for meter's value

Default font      The global default font
Tiny font         A small font (about 6-8 pixels)
Large font        A large font (about 15-36 pixels)
Label font        Used for meter's label
Value font        Used for meter's value
Title font        Used for window titles
Meter title font  Used for meter titles

```

Local timezone must be set if you want to use the clock to display time on some other country or city on earth. You can either obtain the timezone From locale or Set... it by giving the difference

to Greenwich mean time in hours. The timezone offset is negative west of Greenwich, so it's the time that must be added to GMT to obtain the local time. For example, for New York this is -5. Set the Daylight savings option if necessary.

There are built-in lists of countries and cities with timezones in Dashboard. For places not in these lists you'll have to manually enter the timezone offset. If you have Internet access, you should find timezone information from these sites:

```
http://www.hilink.com.au/times/
http://tycho.usno.navy.mil/tzones.html
http://www.bsdi.com/date
http://poisson.ecse.rpi.edu/cgi-bin/tzconvert
```

With the two options Load global settings and Save global settings in Project->Options menu you can control if global settings should be loaded and saved.

## 1.54 Dashboard - Windows

### DASHBOARD - WINDOWS

~~~~~

Windows can be saved and loaded individually.

The Project->Iconify command closes all windows and puts up an icon to the Workbench screen. Double-clicking this icon reopens the windows. The Project->Hide command works like the Hide-button in the Exchange-program. You can reopen the windows with the hotkey set using the CX\_POPKEY option.

When you close a window either by clicking the close gadget or selecting Window->Close window the window is deleted, or if the Project->Options->Close gadget stores window option is set, stored. Stored windows can be found in the Window->Reopen old window menu. Window->Delete window will always close and delete the window.

When the last window is closed, the program will exit. If the Project->Options->Confirm quit option is set, you'll be asked if you really want to close the last window.

The Window->Options->Backdrop window option will make the window a backdrop window, i.e. it will be behind all other windows. A backdrop window can't be resized or moved with the standard window dragbar and size gadget, but when you change the window frame to something else, you'll get invisible window gadgets that can move and size even a backdrop window. More about this later.

Normally, meters are automatically resized when the window size is adjusted. Clear the Rescale meters on window resize option if you want to retain the meter sizes.

The Stay front option will keep the window on top of all other windows.

Dashboard must redraw its windows quite often when you change something. Sometimes this can become frustrating, especially if you're doing a lot of changes at once. Set the Don't redraw option to prevent the window from being redrawn, although it's still sometimes necessary for Dashboard to redraw the window.

When the window is on a screen with aspect ratio approximately 2:1, Dashboard will draw vertical lines with double width so they'll look as thick as horizontal lines. You can control this with the Automatic aspect ratio and 2:1 aspect ratio options.

Dashboard can open its windows on any public screen. You can set the public screen name using the Window->Public screen->Name... menu item. The Jump to next pubscreen command makes the window jump to the next public screen. If a window can't be opened on the named public screen, it will be opened on the default public screen.

Window name... sets the window name, obviously which is displayed in the window title bar and also used to identify closed windows.

There are two frames for a window: Window frame is the outer window frame, and Frame is the innermost one.

If you're using the standard Intuition window borders, you can choose from the Window gadgets menu if titlebar and size gadget should be displayed and if the size gadget should be in the bottom or in the right border. If some other frame is used, you'll get invisible window close, drag, zoom, depth and size gadgets. Various sizes are available in the Window gadgets sub-menu. If the Full window size dragbar option is set, the dragbar will be almost the size of the window, but then you can't select/move/resize any meters.

To help you keep the meters aligned, the window has a grid which can be turned on or off from the Grid menu. You can also choose the grid size from there.

When you're moving a meter that is not currently snapped to the grid, it may be impossible to align it with meters that ARE currently aligned with the grid. The solution is to first move the meter to the upper left corner of the window and then move it to desired position. The grid origin is at the upper left corner.

The Re-initialize command is used to re-initialize and redraw everything in the window. This can be useful with the SANA-II meter if a SANA-II device can't be opened before a connection is made. When you issue this command, the SANA-II meter will try to reopen the device. You'll also need this if you start VMM after Dashboard and you have a meter displaying available memory in some window.

By default, the colors and fonts in the second Window menu point to the global default colors.

## 1.55 Dashboard - Meters

---

## DASHBOARD - METERS

~~~~~

Here's a list of available meters:

Global

CPU Usage

Recent CPU Usage

Load average

Total used CPU time

Total idle CPU time

Context switches

Tasks running/sleeping

Task

CPU Usage

Recent CPU Usage

Used CPU time

Used CPU time (+children)

Child CPU time

Context switches

Stack

Storage

Memory

Disk usage

Device I/O

Drive light

VMM stats

Misc

Clock

Date

Calendar

On-line

SANA-II status

---

SANA-II status (LED)

Textbox

Uptime

Users

To create a meter, choose one of the meters from the Meters menu and using your mouse, draw a box somewhere on the window. The meter will appear. If you'll just get an empty frame, the meter is too small, so it can't be drawn. Enlarge the meter by grabbing one of the corners or edges and dragging it to the desired position. You can also move the meter with mouse.

To select a meter, just click on it. If the window is not active, the first click on the window only activates the window, unless you clear the Project->Options->Eat first click option.

To select multiple meters, hold down the SHIFT-key. To unselect all selected meters, click somewhere in the window.

You can also "lasso" multiple meters by drawing a box on top of the meters to be selected.

By holding down the SHIFT-key, you can move and resize all the selected meters at once.

When Dashboard expects some input from the mouse, the mouse pointer will change to white crosshair. If you don't like this, turn off the Project->Options->Use custom pointer option.

Use the right mouse button to cancel an action like move, resize, adding a new meter, etc.

You can delete a meter with Meters->Delete meter(s) command, or store the meter for later use with Store meter(s) command. Stored meters appear in the Stored meters sub-menu. When you select one of these meters, you place it into the window like a new meter.

You can Cut or Copy meters and Paste them to the same or to another window. If multiple meters are "pasted", they'll be moved to the Stored meters sub-menu, otherwise you'll just place the meter into the window.

The Set aspect command examines the current screenmode aspect ratio, so 1:1 should always be square, no matter what screen resolution is used.

It's possible to place to meters so that they overlap, although Dashboard can't draw overlapping meters correctly. The need for the Put behind others command is explained in the

Tips & tricks

-

section.

When multiple meters are selected, you can use the Align,

Clone size, Set spacing and Spread commands to modify them. To Spread the meters, choose either Horizontal or Vertical from the sub-menu and then click on the left or top edge of the area where you want the meters to be spread and drag the another line to the desired position.

For the Align, Clone size, Set spacing commands the order in which the meters have been selected is important. Meters are aligned with the FIRST selected meter and size will also be cloned from the first selected meter. The Set spacing command will move the meters to the left or below of the first selected meter. The Select all command should not be used to select meters for these commands, because you can't define what meter should be the first selected one.

When one meter is selected, two new menus appear. The first menu contains things that are common to all meters and menus that are used to control what the current meter should display. The second menu is used to set the meter type (bar, text, round, graph, etc.) and control type specific settings.

From the first menu you can adjust the meter frame and fill. You can also copy and paste frames and fills to other meters.

For most of the meters you can set an update interval. The smallest update interval is 1/20 seconds, but this is not available for all meters, because Executive updates most of its values (like load averages) only once per second. CPU usage can be updated four times per second. Only the available update intervals will appear in the Update sub-menu.

If you set the Options->No position or size adjustment option, you can't move or resize the meter. The Rescale on window resize option is similar to the global Rescale meters on window resize command in the Window->Options sub-menu, it only affects this meter.

Each meter has a name which can be changed with the Name... menu item. The name is used to identify stored meters and it's displayed with the title-frame type.

Sometimes it's necessary to set the Meter position inside the meter frame. For example, you may want to have multiple clocks:

New York	11:42
London	16:42
Helsinki	18:42

The space required by the country name is first taken from the meter area and then the time is centered to the remaining space. Because the names are of different length, the times will not be aligned properly. You must align each meter to right by selecting Clock->Meter position->Right.

Each meter has a label which can be automatic or a custom string set by the user. Some meters create an automatic label, for example, the memory-meter displays memory type in the label if Label->Automatic

---



label option is set. Otherwise meter name is displayed in the label, but using the Label->Label... this string can be edited. Using the options in the Label-menu, you can position the label almost anywhere you want, and change the label color and font.

Some meters can display a numerical value or percentage as well as the actual meter. There will be a sub-menu under the Label-menu for controlling this. You can position the value in the same way as a label.

The first item in the second meter-specific menu is a Type sub-menu which is used to choose the meter type (bar, graph, etc.). For some meters you'll also find an option Square (aspect 1:1) from this menu. This is used to keep round meters circular.

Meters that display an unlimited value, e.g. the context switch meter, use a variable scale that is controlled from the Scale type sub-menu in this menu. You can set this to Automatic, and Dashboard automatically changes the scale when necessary. Otherwise you can select one of the fixed ranges from the menu.

## 1.56 Dashboard - CPU usage

DASHBOARD - CPU USAGE

~~~~~

Display CPU usage percentage. This meter can be updated four times per second.

TYPES

Graph

Fuel

Bar

Round

Text

LCD

## 1.57 Dashboard - Recent CPU usage

DASHBOARD - RECENT CPU USAGE

~~~~~

Display recent (last minute) CPU usage percentage.

---

## TYPES

Graph

Fuel

Bar

Round

Text

LCD

## 1.58 Dashboard - Load average

## DASHBOARD - LOAD AVERAGE

~~~~~

Display load averages. You can select the period (1, 5 or 15 minutes) from the Load average->Period sub-menu.

Select Load average->Label->Display to display the load average period as text.

## TYPES

Auto graph

Graph

Round

Text

LCD

## 1.59 Dashboard - Total used CPU time

## DASHBOARD - TOTAL USED CPU TIME

~~~~~

Display total used CPU time.

## TYPES

Text

LCD

Analog

## 1.60 Dashboard - Total idle CPU time

DASHBOARD - TOTAL IDLE CPU TIME

~~~~~

Display total idle CPU time.

TYPES

Text

LCD

Analog

## 1.61 Dashboard - Context switches

DASHBOARD - CONTEXT SWITCHES

~~~~~

Display number of context switches, either total or for last second if the Context switches->Last second option is on. From the What-menu you can choose what is displayed:

Involuntary context switches	Happens when task is forced off the CPU when a higher priority task becomes ready, or when round-robin between tasks of same priority occurs.
Voluntary context switches	Happens when task voluntarily gives up the CPU by calling Wait().
Total	Involuntary + voluntary

TYPES

Text

LCD

Graph

Auto graph

Bar

Round

## 1.62 Dashboard - Tasks running/sleeping

DASHBOARD - TASKS RUNNING/SLEEPING

~~~~~

Display total number of tasks, or just tasks running or sleeping.  
Use the What-menu to select what is displayed.

If you turn on the Tasks->Label->Display option,  
"Tasks running", "Tasks sleeping", etc. will be displayed.

TYPES

Text

LCD

Graph

Auto graph

Bar

Round

## 1.63 Dashboard - CPU usage

DASHBOARD - CPU USAGE

~~~~~

Display task's CPU usage percentage.

Use the Task... menu item to select a task, default is the  
Dashboard task. To display the task name, select CPU usage->  
Task name->Display.

TYPES

Graph

Fuel

Bar

Round

Text

---

LCD

## 1.64 Dashboard - Recent CPU usage

DASHBOARD - RECENT CPU USAGE

~~~~~

Display task's recent (last minute) CPU usage percentage.

Use the Task... menu item to select a task, default is the Dashboard task. To display the task name, select Recent CPU usage-> Task name->Display.

TYPES

Graph

Fuel

Bar

Round

Text

LCD

## 1.65 Dashboard - Used CPU time

DASHBOARD - USED CPU TIME

~~~~~

Display used CPU time.

Use the Task... menu item to select a task, default is the Dashboard task. To display the task name, select Used CPU time-> Task name->Display.

TYPES

Text

LCD

Analog

## 1.66 Dashboard - Used CPU time (+children)

---

---

 DASHBOARD - USED CPU TIME (+CHILDREN)
 

---

Display CPU time used by task and its exited childtasks.

Use the Task... menu item to select a task, default is the Dashboard task. To display the task name, select Used CPU time (+children)->Task name->Display.

## TYPES

Text

LCD

Analog

## 1.67 Dashboard - Child CPU

---

 DASHBOARD - CHILD CPU
 

---

Display CPU time used by task's exited childtasks.

Use the Task... menu item to select a task, default is the Dashboard task. To display the task name, select Child CPU->Task name->Display.

## TYPES

Text

LCD

Analog

## 1.68 Dashboard - Context switches

---

 DASHBOARD - CONTEXT SWITCHES
 

---

Display number of context switches, either total or for last second if the Context switches->Last second option is on. From the What-menu you can choose what is displayed:

Involuntary context switches	Happens when task is forced off the CPU when a higher priority task becomes ready, or when round-robin
------------------------------	--

---

	between tasks of same priority occurs.
Voluntary context switches	Happens when task voluntarily gives up the CPU by calling Wait().
Total	Involuntary + voluntary

Use the Task... menu item to select a task, default is the Dashboard task. To display the task name, select Context switches->Task name->Display.

#### TYPES

Text  
 LCD  
 Graph  
 Auto graph  
 Bar  
 Round

## 1.69 Dashboard - Stack

### DASHBOARD - STACK

~~~~~

Display stack usage as text or percentage.

Use the Task... menu item to select a task, default is the Dashboard task. To display the task name, select Stack->Task name->Display.

#### TYPES

Text  
 LCD  
 Fuel  
 Graph  
 Bar  
 Round

---

## 1.70 Dashboard - Memory

### DASHBOARD - MEMORY

---

Display amount of free memory either as text or percentage. Use the What-menu to select what is displayed. For chip, fast and virtual memory types you can also set the Largest option and the largest free block will be displayed.

VMM and GigaMem are supported. If you start VMM after this meter has been created, you'll have to use the Window->Re-initialize command, because Dashboard doesn't automatically notice when VMM is started. If you're using GigaMem, use the VMEM option when starting Dashboard.

If you turn on the Memory->Label->Display option, the memory type will be displayed.

#### TYPES

Bar  
Fuel  
Graph  
Round  
Text  
LCD

## 1.71 Dashboard - Disk usage

### DASHBOARD - DISK USAGE

---

Display disk usage, how much of the disk is currently being used. Use the Drive... menu item to select the drive. Both devices and volumes can be chosen. Text and LCD meters display amount of free disk space.

If you turn on the Disk usage->Label->Display option, drive name will be displayed.

The default drive is "RAM:".

#### TYPES

Bar  
Fuel

---



Graph

Round

Text

LCD

## 1.72 Dashboard - Device I/O

DASHBOARD - DEVICE I/O

~~~~~

Display amount of device I/O, reads and/or writes. You can choose to display either total or the current amount. Use the Device... menu item to select the device. Note that only full devices can be displayed, it's not possible to distinguish I/O from/to different device units or partitions.

You can monitor a serial device with this meter, but not AmiTCP, as it seems to always read approximately 2xMTU bytes at once. Dashboard doesn't open the monitored device, so if the serial-device isn't open when this meter is created, you'll have to use the Window->Re-initialize command later, when the device has been opened.

When this meter is used, Dashboard will patch exec.library DoIO() and SendIO() functions. These patches can't be removed when Dashboard exits if somebody is still executing these functions. Dashboard will notify you about this with a requester and then just keep trying until the patches can be removed. PatchControl is supported.

If you turn on the Device I/O->Label->Display option, device name will be displayed.

Device names are case sensitive.

The default device is "trackdisk.device", unit 0.

TYPES

Graph

Auto graph

Bar

Round

Text

LCD

## 1.73 Dashboard - Drive light

### DASHBOARD - DRIVE LIGHT

---

Display a LED for a device. The LED will blink when data is read and/or written from/to the disk. Use the Device... menu item to select the device.

With this meter you can also create read and write LEDs for you modem, just select the serial device you're using.

If you turn on the Drive light->Label->Display option, the device name will be displayed.

Device names are case sensitive.

The default device is "trackdisk.device", unit 0.

#### TYPES

LED

## 1.74 Dashboard - VMM stats

### DASHBOARD - VMM STATS

---

Display VMM statistics. Different values are available in the What sub-menu. Current value can't be displayed for Pageframes and Pages on dev values. For others, either a total or the current value can be displayed.

Instead of displaying number of pageframes, size in kilo- or megabytes is displayed.

Use the

Memory  
meter to display the amount of free virtual  
memory.

If you turn on the VMM stats->Label->Display option, the name of the value currently shown will be displayed.

#### TYPES

Graph

Auto graph

Bar

Round

---

Text

LCD

## 1.75 Dashboard - Clock

DASHBOARD - CLOCK

~~~~~

Display a clock. If you want to display time for some other location on earth, you'll first have to set the local timezone from the Global->Local timezone menu.

You can either set the timezone using the Clock->Timezone menu (see

Global settings

-section for this), or choose a country or a city.

The Local time option switches between local time or the timezone you have set.

TYPES

Analog

Text

LCD

## 1.76 Dashboard - Date

DASHBOARD - DATE

~~~~~

Display current date. You can also choose a timezone, like in the

Clock  
meter.

TYPES

Text

LCD

## 1.77 Dashboard - Calendar

---

## DASHBOARD - CALENDAR

~~~~~

Display a simple calendar. You can also choose a timezone, like in the

Clock  
meter.

You can choose a separate font for day number, displayed in the middle, and for day- and month names displayed at top and bottom.

## 1.78 Dashboard - On-line

## DASHBOARD - ON-LINE

~~~~~

Display how long your modem has been on-line. Set the serial device and unit using the Device... and Device unit... menus.

This meter should work with all serial devices, but your modem cable must have a DCD-wire for carrier detect-signal. The timer starts when carrier is detected and stops when carrier is lost. This meter will of course work with AmiTCP.

-----

Your communication program must open the serial device in SHARED mode. If you're using AmiTCP, add "SHARED" to the S2OPTS-variable in bin/startnet-script.

-----

Device names are case sensitive.

The default device is "serial.device".

TYPES

Text

LCD

Analog

## 1.79 Dashboard - SANA-II status

## DASHBOARD - SANA-II STATUS

~~~~~

Display SANA-II device (PPP, a2065, aslip) status. Various values are available in the What-menu:

|                  |                                                                                               |
|------------------|-----------------------------------------------------------------------------------------------|
| Packets received | Number of packets that the unit has received.                                                 |
| Packets sent     | Number of packets that the unit has sent.                                                     |
| Bad packets      | Number of bad packets received (i.e., hardware CRC failed).                                   |
| Packets dropped  | Number of packets dropped due to insufficient resources available in the network interface.   |
| Unknown packets  | Number of packets received that had no pending read command with the appropriate packet type. |
| Reconfigurations | Number of network reconfigurations since the unit was last configured.                        |

Either a total or current value can be displayed.

Set the SANA-II device and unit using the Device... and Device unit... menus. The name is common to ALL SANA-II meters (including LEDs), so you'll have to set it from one meter only.

If the device is in the DEVS: directory, only the device name must given, if it's in the DEVS:networks/ directory, then a full path is required.

Device names are case sensitive.

If you turn on the SANA-II status->Label->Display option, the name of the value currently shown will be displayed.

#### TYPES

Graph

Auto graph

Bar

Round

Text

LCD

## 1.80 Dashboard - SANA-II status (LED)

DASHBOARD - SANA-II STATUS (LED)

~~~~~  
Display SANA-II device status LED.

See

SANA-II status  
meter for more information.

TYPES

LED

## 1.81 Dashboard - Textbox

DASHBOARD - TEXTBOX

~~~~~

Display a simple box with optional text. From the second Textbox menu you can set the string that is displayed, turn the text on or off and set font and color that are used for the text.

## 1.82 Dashboard - Uptime

DASHBOARD - UPTIME

~~~~~

Display system uptime.

TYPES

Text

LCD

Analog

## 1.83 Dashboard - Users

DASHBOARD - USERS

~~~~~

Display number of users. MultiUser must be installed.

TYPES

Graph

Auto graph

Bar

---

Round

Text

LCD

## 1.84 Dashboard - Round

DASHBOARD - ROUND

~~~~~

The round meter type consists of a face,  
 scale  
 , hands and a centre.

The face can be round, oval or rectangle. It can be chosen from the Round->Face menu. The other options in this menu are:

Display	face on/off
Solid	solid face
Edges	draw edges
Bevelled	bevelled edges
Inverted	invert the edge colors

A centre is drawn on top of the needles, it can be turned on or off from the Round->Centre menu. Its color can be changed from this menu also.

For each hand you get two menus. From the first menu you can turn the hand on or off and choose between wireframe and solid polygon. Several hand types are available in the second menu.

## 1.85 Dashboard - Analog

DASHBOARD - ANALOG

~~~~~

This is used to display a clock, or some other time. It's like the

Round  
 meter type, except that you can choose roman numbers instead of the "1", "2", "3", etc.

## 1.86 Dashboard - Graph

---

 DASHBOARD - GRAPH
 

---

Graph consists of two parts: a scrolling graph and a scale

.

A frame is drawn around the graph and the graph has a solid background. These can be changed from the Graph menu.

The graph can also be a histogram, this can be changed from the Graph->Graph submenu. From there you can also choose if scalelines are displayed and change the graph and scaleline color.

You can set the scroll speed from the Graph->Scroll menu, 1/2 means that the graph is scrolled half the width of the graph.

The Graph->Clear clears the graph.

## 1.87 Dashboard - Auto graph

---

 DASHBOARD - AUTO GRAPH
 

---

Similar to

graph

meter type, but no scale is displayed. Instead, the graph is automatically scaled as the value varies, by drawing a number of horizontal lines across the graph. Each line represents an increment of a user defined value. This value can be changed with the Auto graph->Scale division... command. For load average the value is fixed to 1.0. So, to display load average 2.40, the graph is divided in three parts and two horizontal lines are drawn to indicate values 1.0 and 2.0.

You can set the minimum number of scalelines using the Graph->Graph->Minimum # of scalelines command.

## 1.88 Dashboard - Bar

---

 DASHBOARD - BAR
 

---

Bar can be either horizontal or vertical. A scale is drawn for the bar.

You can set the bar color from the Bar->Bar menu, where you'll also find an option to turn the solid graph to blocks.

---



## 1.89 Dashboard - Fuel

DASHBOARD - FUEL

~~~~~

This tries to mimic a car fuel meter. You can adjust colors and a font used for the "E" and "F" from the Fuel-menu.

## 1.90 Dashboard - LED

DASHBOARD - LED

~~~~~

LED is either circular or rectangular, or it can be a user defined text.

Use the LED->Text... menu item to define a text string that is displayed. The LED->Text option will be automatically turned on.

You can also define colors for ON and OFF states from the LED-menu.

Dashboard must get some CPU time to update the LED.

## 1.91 Dashboard - Text

DASHBOARD - TEXT

~~~~~

This displays the value as text. Text color and font can be changed from the Text-menu. A frame can be drawn around the text, and the frame area can be filled.

## 1.92 Dashboard - Text

DASHBOARD - TEXT

~~~~~

This displays the value as text. Text color and font can be changed from the Text-menu. A frame can be drawn around the text, and the frame area can be filled.

Use the Text->Format sub-menu to choose if the value should be displayed in bytes, kilobytes or megabytes, or choose Automatic.

The Percentage, Display memory type and Display 'free' options turn on or off various parts of the text. The last two options are available only for the free memory meter.

---

### 1.93 Dashboard - Text

DASHBOARD - TEXT

~~~~~

This displays the value as text. Text color and font can be changed from the Text-menu. A frame can be drawn around the text, and the frame area can be filled.

The Text->Short option, when on, displays the value in kilos and megabytes.

### 1.94 Dashboard - Text

DASHBOARD - TEXT

~~~~~

This displays the time as text. Text color and font can be changed from the Text-menu. A frame can be drawn around the text, and the frame area can be filled.

Short, Compact and Long formats are available.

For some time values there's also an option to display 1/100 seconds.

### 1.95 Dashboard - Text

DASHBOARD - TEXT

~~~~~

This displays the time as text. Text color and font can be changed from the Text-menu. A frame can be drawn around the text, and the frame area can be filled.

Time can be displayed in AmigaOS format, in locale format, or in user defined format. Only the AmigaOS format can be used in OS2.04, at least OS2.1 is required for the others.

When you select the User defined... menu item, you can enter a format string, which is used to construct the time/date string. The default format string for clock is "%H:%M:%S" and for date "%Y-%m-%d". The string may contain any other characters too, not just commands. For example:

```
"It's %A, %B %e, %Y."
```

will produce:

```
It's Tuesday, July 9, 1996.
```

Note that although it's possible to display both time and date using either a clock or date meter, the date meter is not updated every

---

second.

The following commands are supported:

```
%a - abbreviated weekday name
%A - weekday name
%b - abbreviated month name
%B - month name
%c - same as "%a %b %d %H:%M:%S %Y"
%C - same as "%a %b %e %T %Z %Y"
%d - day number with leading 0s
%D - same as "%m/%d/%y"
%e - day number with leading spaces
%h - abbreviated month name
%H - hour using 24-hour style with leading 0s
%I - hour using 12-hour style with leading 0s
%j - julian date
%m - month number with leading 0s
%M - the number of minutes with leading 0s
%n - insert a linefeed
%p - AM or PM strings
%q - hour using 24-hour style
%Q - hour using 12-hour style
%r - same as "%I:%M:%S %p"
%R - same as "%H:%M"
%S - number of seconds with leadings 0s
%t - insert a tab character
%T - same as "%H:%M:%S"
%U - week number, taking Sunday as first day of week
%w - weekday number
%W - week number, taking Monday as first day of week
%x - same as "%m/%d/%y"
%X - same as "%H:%M:%S"
%y - year using two digits with leading 0s
%Y - year using four digits with leading 0s
```

## 1.96 Dashboard - Scale

DASHBOARD - SCALE

~~~~~

Similar scale is used for multiple meter types, namely graph, bar and round types.

You'll find the scale menus under the === SCALES === label. The scale consists of "ticks" and values.

The Display option is used to toggle the scale on or off. For round type meters you can choose if the scale values are inside or outside using the Scale inside option.

From the Value font menu you can choose a font for the values. By default this points to the window's "Tiny font".

The scale is be divided into maximum of four groups. For example,

an analog clock is divided to noon, quarter, hour and minute groups. They can all be individually controlled. You'll find two sub-menus for each of the four groups. The first menu contains the following items:

```
V Display
V Bevelled
V Values
~~~~~
=== COLOR 1 ===
...
~~~~~
=== COLOR 2 ===
...
~~~~~
=== FILL COLOR ===
...
~~~~~
=== VALUE COLOR ===
```

The Display option used to turn the specific scalegroup on or off. If Bevelled is not set, the "ticks" will be drawn using only color 1 and fill color. You can also individually turn Values on or off for each group.

Color 1, color 2 and fill color are used when drawing the "ticks". Values are rendered in the value color.

From the second menu, e.g. "Noon type", you can choose the "tick" type.

## 1.97 Dashboard - ScreenNotify

DASHBOARD - SCREENNOTIFY

~~~~~

Dashboard supports Stefan Becker's ScreenNotify. To make it work, just copy the screennotify.library to LIBS:.

When ScreenNotify is in use, Dashboard will close its windows from Workbench screen or any public screen that is being closed. When the screen is reopened, Dashboard will reopen the closed windows. This lets you change, for example, Workbench screenmode or fonts while Dashboard is running.

Dashboard will disable ScreenNotify if it must open a requester, because otherwise you couldn't open or close a screen while the requester is being displayed.

If you don't have ScreenNotify, you can FTP it from any Aminet site, for example ftp.wustl.edu. The file is:

```
util/libs/ScreenNotify10.lha
```

## 1.98 Dashboard - Tips & tricks

### DASHBOARD - TIPS & TRICKS

~~~~~

Here are some more or less useful tips and tricks.

#### Multiple fills

You can fill a window using the dither fill type, but it can only handle two colors. If you want to create a "horizon" with graduated sky and ground, you need four colors. Here's what you can do:

- Create two Textbox meters:

```

|-----|
|   |   | <- window
|   |   |
|   |   |
|-----|
|   |   |
|   |   |
|-----|

```

The first Textbox should occupy about 3/4 of the height of window.

- Set the Textbox->Options->No position or size adjustment option, so you don't accidentally move the meters
- Set Textbox->Frame to None
- Move both meters behind all others with Textbox->Put behind others
- Set Textbox->Fill to Dither
- Set the colors (RGB) for both fills:

```

top      color 1    0 15 90
          color 2   30 75 180
bottom   color 1  105 30  0
          color 2  150 90  45

```

- Unselect the meters using Meters->Unselect all

#### Bevelled lines

You can add bevelled lines to windows by creating a Textbox meter and turning off all other edges of the frame except one. Set frame type to Double.

#### Images

If you want to add an image somewhere on a Dashboard window, create a Textbox meter, and set Textbox->Fill to Image....

#### Modem read and write LEDs

You can use the

Drive light  
meter to create read and write LEDs  
for your modem.

Icon for preference files

The "Icon"-file in DashboardPrefs directory is the icon file that  
will be copied for saved preference files.

## 1.99 Dashboard options

### DASHBOARD OPTIONS

~~~~~  
FROM

Template: FROM/K  
Tooltype: FROM  
Default: "Default" (the default preferences file)

The initial preference file.

PREFSDIR

Template: PREFSDIR/K  
Tooltype: PREFSDIR  
Default: PROGDIR:DashboardPrefs

Directory where preference files are stored.

CX\_PRIORITY

Template: CX\_PRIORITY/N/K  
Tooltype: CX\_PRIORITY  
Default: 0

Commodity priority (from -128 to 127).

CX\_POPUP

Template: CX\_POPUP/K  
Tooltype: CX\_POPUP  
Default: yes

This can be "yes" or "no". If it's "no", then Dashboard will not open  
its window(s) when it is started.

CX\_POPKEY

Template: CX\_POPKEY/K  
Tooltype: CX\_POPKEY  
Default: no hotkey

---

The hotkey which will open Dashboard windows closed with the Project->Hide or Project->Iconify command.  
For example, "ctrl alt d".

#### NOSCREENNOTIFY

Template: NOSCREENNOTIFY/S  
Tooltype: NOSCREENNOTIFY

Don't use  
                  screennotify.library  
                  .

#### VMEM

Template: VMEM/S  
Tooltype: VMEM

If you're are using GigaMem, specify this option. By default the "vmem.library" is not opened, because there are many libraries with the same name, and only the GigaMem's "vmem.library" can be used.

#### HELP

Template: HELP/S  
Tooltype: HELP

Display this document.

#### VERSION

Template: VERSION/S  
Tooltype: -

Display version number.

## 1.100 ExecutivePrefs

### EXECUTIVEPREFS

~~~~~

Change server preferences.

#### OPTIONS DESCRIPTION

Two versions of ExecutivePrefs are provided, a GadTools-version for all Amigas and one which supports MUI.

ExecutivePrefs searches the preferences file from these directories, in this order:

```

current directory
PROGDIR: (directory where the ExecutivePrefs binary is)
ENV:
S:

```

If the file can't be loaded, default settings are used.

If you want to move ExecutivePrefs to SYS:Prefs directory, you must also move the Executive.prefs file from Executive-directory to ENVARC: so ExecutivePrefs will find it.

The Save-button saves the preferences back to the file that it loaded, or to "PROGDIR:Executive.prefs"-file if no preferences file was loaded. The changes won't take effect before the server is restarted.

The Save/Restart-button saves the preferences file and restarts the server. You'll lose information about window owners and task relationships if you restart the server.

ExecutivePrefs is divided into four sections:

```

Tasks
Options
Ranges
Scheduler

```

The Tasks-section lets you add a task to a list of watched tasks. A list of tasks currently being watched is displayed, for example:

```

TYPE          NAME          ACTION          PRIORITY    NICE
TASK          Top           SCHEDULE        0            -15
CHILDTASKS   ASwarm       NOSCHEDULE      BELOW
IGNORE       BuggyProgram*
...

```

The TYPE-field is TASK, CHILDTASKS, TASK+CHILDTASKS or IGNORE. TASK affects the named task only, CHILDTASKS doesn't affect the named task, but all its childevents. TASK+CHILDTASKS affects the named task and its childevents. If TYPE is IGNORE, then the named task will be completely ignored. The IGNORE option is VERY rarely needed with tasks that aren't removed from the system in a legal way, i.e. calling exec.library RemTask() function.

NAME is the task name. Ps, Top and some other clients might display a task name in square brackets, which means that it's a shell command. The brackets don't actually belong to the task name, so don't use the brackets here.

TASK, CHILDTASKS and TASK+CHILDTASKS requires that the ACTION is also specified. ACTION-field is SCHEDULE, NOSCHEDULE or RELATIVE. SCHEDULE forces a task to be scheduled, NOSCHEDULE tells the server NOT to schedule the task, even if its priority would be in the catch range. RELATIVE only works together with CHILDTASKS, it's explained here

.



PRIORITY is used only with NOSCHEDULE. If it's blank, nothing will be done to task's priority. If it's ABOVE, the priority will be kept above the dynamic range (>DYNAMICMAX). BELOW will keep the priority below the dynamic range (<DYNAMICMIN). If it's a number, then the affected task(s) will be given this priority.

NICE is used only with SCHEDULE, it lets you set a default nice-value for the task. If no nice-value is specified here, then the parent task's nice-value is used.

See the  
server documentation  
for more information.

IMPORTANT! For Executive to catch a task, you must start it from a shell with the Run-command, or start it from Workbench. If you just execute the command in shell, no new task is created and Executive won't notice the command.

The New-button creates a new entry and opens the edit-window, which can also be opened by double-clicking a task, or pressing the Edit-button.

The Copy-button makes a copy of the selected task and the Delete-button deletes the selected task.

The Scan-button scans the system for tasks in the Magic Wand-list.

The Edit-window is used to adjust settings for the selected entry. You can change the name of the task, and as said earlier, \* is allowed as a wildcard. By pressing the popup-button in the MUI version of ExecutivePrefs or the Name-button in the GadTools version, you can choose a name from a list of tasks currently in running.

The \* MAGIC WAND \* -button opens a list of preset entries for certain programs. When you select a task from this list, the settings will be adjusted and you can just press the Ok-button to confirm. Sometimes you might need to give a name to the task, if it's not specified in the Magic Wand.

I have included several common screenblankers, multi-function commodities and communication programs to the Magic Wand list, if you know other programs that should on the list, please let me know.

The explanation window reflects the state of the buttons below it. It will explain what will happen to the task. Try different settings and see how it changes.

If task is set to be scheduled, you can give task a default nice-value or use parent task's nice-value. If you want to specify an initial nice-value for a task, click the checkbox to unghost the nice-value slider.

Please read the  
Server internals  
and  
Common problems

---

sections for  
more information.

The Options-section is used to set a couple of miscellaneous options:

"No focus on active window" disables  
focus  
on active window.

"Schedule Workbench" will  
schedule the Workbench task  
. By default  
Workbench is not scheduled.

"Don't fix OpenWorkbench bug" tells the server not to fix a  
bug  
in AmigaOS 3.x.

"Power LED shows CPU usage" lets you to monitor CPU usage with the  
power LED. Because turning off the power LED also turns off the low-  
pass audio filter, there will be some interference in audio output.

You can also change the preferred  
timer  
from here.

The Ranges-section lets you adjust the  
catch and dynamic ranges

.  
ExecutivePrefs makes sure that you don't specify illegal values here.  
It permits you to set a high maximum priority for both ranges, but try  
to avoid setting the catch range maximum value over 4.

The Scheduler-section lets you choose a  
scheduler

.  
Remember that the changes you make don't affect the currently running  
server, you must restart it.

The Settings->MUI Preferences... opens then MUI preferences editor.

## EXAMPLES

ExecutivePrefs

## 1.101 ExecutivePrefs options

EXECUTIVEPREFS OPTIONS

~~~~~

---

## FROM

Template: FROM  
Tooltype: FROM

Name and path of the preferences file if it isn't in any of the directories where ExecutivePrefs looks for it first.

## LEFT

Template: LEFT/N/K  
Tooltype: LEFT

Offset of the left edge of the window relative to screen. This option is not implemented in the MUI-version of ExecutivePrefs.

## TOP

Template: TOP/N/K  
Tooltype: TOP

Offset of the top edge of the window relative to screen. This option is not implemented in the MUI-version of ExecutivePrefs.

## WRITEDEFAULTS

Template: WRITEDEFAULTS/K  
Tooltype: -

The installation script uses this option to write a default preferences file.

## UPDATEPREFS

Template: UPDATEPREFS/S  
Tooltype: -

The installation script uses this option to update existing preferences file.

## ADD

Template: ADD/K  
Tooltype: -

The installation script uses this option when generating/updating a preferences file.

## HELP

Template: HELP/S  
Tooltype: HELP

Display this document.

## VERSION

Template: VERSION/S

Tooltype: -

Display version number.

## 1.102 Kill

### KILL

~~~~~

Send a break-signal to task.

### OPTIONS DESCRIPTION

Kill is similar to the standard Amiga utility Break, but it has a lot more options. Kill sends a break signal to specified tasks but it's not guaranteed that these tasks will react to the signal.

Where Break can only send break-signal to one task at a time, Kill can handle several tasks with one command. For example:

```
Kill PID=10,20 NAME=DH\* USER=root GROUP=staff PGRP=273
```

The following tasks will be affected by this command:

(NOTE: Depending on the shell you use, the \ may not be needed in front of the asterisk.)

- tasks with PID 10 and 20
- tasks with name starting with "DH" (case insensitive).
- all tasks owned by the root user. (MultiUser only)
- all tasks owned by the staff group. (MultiUser only)
- all tasks that belong to process group 273.

There's no need to use the PID-keyword, Kill can be invoked without it too:

```
Kill 10
```

This will send break-signal to task with PID 10.

You can replace c:Break with Kill, they are compatible and Kill doesn't need the server to run. If the server isn't running you can use only PIDs that correspond to

CLI numbers

.

You can't kill the server with Kill, because Kill connects to the server and the server can't exit if it gets a break-signal because Kill has connected to it.

The PGRP option is really useful when you, for example, want to kill

a GNU C compilation. "Pstree SHOWPIDS" outputs:

```

`---[Make] (9)
  +---CON (236)
    `---[sh] (1)
      `---[cpp] (10)

```

You can send a break-signal to all these tasks with a single command:

```
Kill PGRP=9
```

If MultiUser has been installed, then root privileges are needed to kill other users' tasks. If you don't want to require root privileges for Kill, set the U-bit using MProtect.

#### EXAMPLES

Sorry, it's not possible to give any good examples of Kill.

## 1.103 Kill options

### KILL OPTIONS

#### PID

```

Template:  PID=PROCESS
Tooltype:  PID

```

PIDs of the tasks that will be killed. You can list multiple PIDs here, separate them with a comma, for example "PID=10,248,332".

#### NAME

```

Template:  N=NAME/K
Tooltype:  NAME

```

Names of the tasks that will be killed. You can list multiple names here, separate them with a comma, for example "NAME=DH0,DH1,DH2,Shell\*". You can use \* as a wildcard.

#### USER

```

Template:  U=USER/K
Tooltype:  USER

```

Names of the users whose all tasks will be killed. You can list multiple user names here, separate them with a comma, for example "USER=Frodo,Pippin,Gandalf". You can also use UIDs. This option isn't available if MultiUser hasn't been installed.

#### GROUP

Template: G=GROUP/K  
Tooltype: GROUP

Names of the groups whose all tasks will be killed. You can list multiple group names here, separate them with a comma, for example "NAME=staff,students". You can also use GIDs. This option isn't available if MultiUser hasn't been installed.

#### PGRP

Template: R=PGRP/K  
Tooltype: PGRP

PGRPs of the process groups whose all tasks will be killed. You can list multiple PGRPs here, separate them with a comma, for example "PGRP=254,298,578".

#### ALL

Template: ALL/S  
Tooltype: ALL

Send all signals (CDEF) to the task(s).

#### C

Template: C/S  
Tooltype: C

Send the C-signal to the task(s). This is the default and most commonly used.

#### D

Template: D/S  
Tooltype: D

Send the D-signal to the task(s).

#### E

Template: E/S  
Tooltype: E

Send the E-signal to the task(s).

#### F

Template: F/S  
Tooltype: F

Send the F-signal to the task(s).

#### REMTASK

Template: REMTASK/S  
Tooltype: REMTASK

---

This will REMOVE the specified task(s) from Exec task-lists with RemTask() call. This is EXTREMELY DANGEROUS and you can easily crash your system if you don't know exactly what you are doing.

#### HELP

Template: HELP/S  
Tooltype: HELP

Display this document.

#### VERSION

Template: VERSION/S  
Tooltype: -

Display version number.

## 1.104 Lastcomm

### LASTCOMM

~~~~~

Show last commands executed in reverse order.

#### OPTIONS DESCRIPTION

Lastcomm gives information on previously executed commands. With no arguments, Lastcomm displays information about all the commands recorded during the current accounting file's lifetime. The most recently terminated command will be shown first.

Lastcomm will read from the default accounting file "s:acct", but you can use the ACCTFILE option or the environmental variable ACCTFILE to specify another name for the accounting file.

The output looks like this:

login	-	--H	0.06s	14-Jun-95	00:04:17
lzx	root	S-H	0.55s	14-Jun-95	00:04:32
lha	-	--H	16.04s	14-Jun-95	00:04:07
gzip	-	--H	3.36s	14-Jun-95	00:03:53
csh	-	--H	0.12s	14-Jun-95	00:03:29
endcli	-	--H	0.00s	14-Jun-95	00:03:29
date	-	---	0.01s	14-Jun-95	00:03:28
stack	-	---	0.01s	14-Jun-95	00:03:28
RX	-	--H	0.03s	14-Jun-95	00:03:05
Sa	-	--H	0.84s	14-Jun-95	00:03:04

For each command, the following are displayed:

- command name

- name of the user who executed the command (requires MultiUser)  
(If MultiUser hasn't been installed, the user name will be -)
- flags:
  - S -> command was executed by super-user
  - W -> command was run from Workbench
  - H -> command was scheduled
- amount of CPU time used by the command (in seconds)
- time the command terminated

Lastcomm doesn't require that the server is running.

Lastcomm asks the Acct-daemon to write all unpurged records to disk so they will be included in the output.

CTRL-C will stop the output.

#### EXAMPLES

Lastcomm

## 1.105 Lastcomm options

### LASTCOMM OPTIONS

~~~~~

#### ACCTFILE

Template: F=ACCTFILE  
Tooltype: ACCTFILE  
Default: s:acct

The accounting file name.

#### PROCS

Template: C=COMMANDS/N/K  
Tooltype: COMMANDS  
Default: all

How many commands to display.

#### NAME

Template: N=NAME/K  
Tooltype: NAME

Only commands matching this name should be displayed. You can use \* as a wildcard.

#### USER

Template: U=USER/K  
Tooltype: USER

---



Display commands executed by this user (name or UID). This option isn't available if MultiUser hasn't been installed.

#### HELP

Template: HELP/S  
Tooltype: HELP

Display this document.

#### VERSION

Template: VERSION/S  
Tooltype: -

Display version number.

## 1.106 Meter

### METER

~~~~~

Display system information.

### OPTIONS DESCRIPTION

Meter displays free memory, load averages, CPU usage, uptime, free disk space and lots of other things in a small window. Meter will display each piece of information on one line. A horizontal bar will be displayed if possible. Information in the window is updated by default every second.

Bars indicate a percentage of a current value from a maximum value. For example, currently free memory from total available memory. For some values, it's not possible to display a bar, because there's no maximum value. Maximum value here means for example total available memory.

It's also possible to display a small graph for values that have a maximum value. Double click a value to toggle the graph on/off.

To make the graphs more readable, a few scale lines will be drawn. One solid line on the bottom and a dotted line in the middle and. For load average values this is different, scale lines are drawn for each load average level, like in ALoad.

You can choose from the Show-menu what information should be displayed. The window will be resized if accordingly. There are a few presets in the Settings->Presets menu:

Default            Show default values

User defined	Show the user defined values
Memory	Show memory usage
CPU usage	Show CPU usage
VMM stats	Show VMM statistics

Using the SHOW option you can specify what information should be displayed. SHOW option takes a list of keywords separated by a comma. A 'G' here indicates that a graph can be drawn for this value. Add a "-" after the value name to display a graph for it.

clock	Display a clock
chip	G Free CHIP memory
fast	G Free FAST memory
virtual	G Virtual memory (GigaMem or VMM)
total	G Total free memory
largest	G Largest free block
load1	G Load average over the last minute
load5	G Load average over the last 5 minutes
load15	G Load average over the last 15 minutes
cpuusage	G CPU usage
recentcpu	G Recent CPU (last minute)
unpurged	Number of unpurged accounting records
acctrecs	Total number of accounting records created
addedtasks	Number of added tasks
terminatedtasks	Number of terminated tasks
tasksrunning	Tasks currently running or ready to run
currentcsw	Number of context switches during last second
currentvocsw	Voluntary context switches during last second
currentivcsw	Involuntary context switches during last second
csw	Total number of context switches
vocsw	Voluntary context switches
ivcsw	Involuntary context switches
users	Users logged in (MultiUser only)
uptime	Uptime
pagefaults	Pagefaults (VMM)
pagesread	Pages read (VMM)
pageswritten	Pages written (VMM)
pageframes	Pageframes (VMM)
pagesused	Pages on dev (VMM)

You can also use device or volume names like DF0:, DH0:, Work: etc. as keywords, free disk space will be displayed for these. If there's no disk in drive or the disk is unreadable, a "-" is displayed here.

To display total free memory, CPU usage, uptime and free space on drive DH0: use this command:

```
Meter SHOW=total,cpuusage,uptime,DH0:
```

The order of the keywords is insignificant, as the values will always be in the same order.

I suggest you use some small font so you can display more information in the window. Use the SPACEHOR and SPACEVER options to add more space between the window and the values. The BETWEEN option controls the space between two values and the BAREXTRA option can be used to specify extra height for bars.

A frame can be drawn for the window using the WINDOWFRAME option, and for individual values with the VALUEFRAMES option. These can be inverted using the INVERTWINDOWFRAME and INVERTVALUEFRAMES options.

Use the options BGCOLOR, BARCOLOR, TEXTCOLOR, NUMBERCOLOR, GRAPHCOLOR and GRIDCOLOR to change background, bar, text, number, graph and grid colors, respectively. Frame colors can be set using the FRAMECOLOR1 and FRAMECOLOR2 options.

If you don't want to display any bars, labels or numbers, use the three menu items in the Settings menu:

```
Settings->Show labels  disable/enable labels
Settings->Show numbers "          numbers
Settings->Show bars    "          bars
```

The menu item Project->New will redraw the display. Project->Flush memory flushes all unused libraries from memory. Settings->Stay front keeps Meter's window in front of all other windows.

The Project->Save config menu item will save current settings to the program's icon. This option is available only when the program is started from Workbench.

The window has invisible drag, close and depth gadgets if the system gadgets have been disabled with NOTITLEBAR option. The centre of the window works as a drag-bar.

To turn the Intuition window borders off, use the BORDERLESS option and to put the window behind all other windows, use the BACKDROP option.

If you're using GigaMem, use the VMEM option when starting Meter.

Meter requires a registered version of Executive.

## EXAMPLES

Meter

With window- and value frames:

```
Meter WINDOWFRAME VALUEFRAMES
```

CPU usage as a graph (+ clock):

```
Meter SHOW=clock,cpuusage- GRAPHHEIGHT=150
```

Use a different font:

```
Meter FONT=courier FONTSIZE=13
```

Output a list of keywords:

---

Meter LIST

## 1.107 Meter options

METER OPTIONS

~~~~~

SHOW

Template: S=SHOW  
Tooltype: SHOW  
Default: CLOCK,CHIP,FAST,TOTAL,LARGEST,LOAD1,CPUUSAGE,RECENTCPU,  
CURRENTCSW,UPTIME

What information Meter should display. Separate keywords with a comma. Order of the keywords is insignificant. Add a "-" after a value name to display a graph for it.

UPDATE

Template: UD=UPDATE/N/K  
Tooltype: UPDATE  
Default: 1

The update interval in seconds.

GRAPHHEIGHT

Template: GH=GRAPHHEIGHT/N/K  
Tooltype: GRAPHHEIGHT  
Default: 2x font height

Height of the graphs in pixels.

BAREXTRA

Template: BE=BAREXTRA/N/K  
Tooltype: BAREXTRA  
Default: 1

Bar height is by default the same as the font height, but you can use this option to make the bars taller. Specifying 1 makes the bar two pixels taller, one extra line above the bar and one below.

BGCOLOR

Template: BGC=BGCOLOR/N/K  
Tooltype: BGOLOR

Background color. (0-255)

BARCOLOR

---

Template: BARC=BARCOLOR/N/K  
Tooltype: BARCOLOR

Bar color. (0-255)

#### TEXTCOLOR

Template: TXTC=TEXTCOLOR/N/K  
Tooltype: TEXTCOLOR

Text color. (0-255)

#### NUMBERCOLOR

Template: NBC=NUMBERCOLOR/N/K  
Tooltype: NUMBERCOLOR

Number color. (0-255)

#### GRAPHCOLOR

Template: GRC=GRAPHCOLOR/N/K  
Tooltype: GRAPHCOLOR

Graph color. (0-255)

#### GRIDCOLOR

Template: GRD=GRIDCOLOR/N/K  
Tooltype: GRIDCOLOR

Color of the graph scale lines. (0-255)

#### FRAMECOLOR1

Template: FC1=FRAMECOLOR1/N/K  
Tooltype: FRAMECOLOR1

Windowframe and valueframe light edge color.

#### FRAMECOLOR2

Template: FC2=FRAMECOLOR2/N/K  
Tooltype: FRAMECOLOR2

Windowframe and valueframe dark edge color.

#### NOLABELS

Template: NOLABELS/S  
Tooltype: NOLABELS

Don't display labels.

#### NONUMBERS

Template: NONUMBERS/S

---

Tooltype: NONUMBERS

Don't display numbers.

#### NOBARS

Template: NOBARS/S

Tooltype: NOBARS

Don't display bars.

#### SCROLL

Template: SCR=SCROLL/N/K

Tooltype: SCROLL

Default: 1

Number of pixels the graphs are scrolled each time.

#### WINDOWFRAME

Template: WF=WINDOWFRAME/S

Tooltype: WINDOWFRAME

Draw a frame for the window.

#### VALUEFRAMES

Template: VF=VALUEFRAMES/S

Tooltype: VALUEFRAMES

Draw frames for values.

#### INVERTWINDOWFRAME

Template: IWF=INVERTWINDOWFRAME/S

Tooltype: INVERTWINDOWFRAME

Invert the window frame.

#### INVERTVALUEFRAMES

Template: IVF=INVERTVALUEFRAMES/S

Tooltype: INVERTVALUEFRAMES

Invert the value frames.

#### SPACEHOR

Template: SH=SPACEHOR/N/K

Tooltype: SPACEHOR

Space between the window left/right edge and the values.

#### SPACEVER

Template: SV=SPACEVER/N/K

Tooltype: SPACEVER

Space between the window top/bottom edge and the values.

#### BETWEEN

Template: BET=BETWEEN/N/K

Tooltype: BETWEEN

Space between values.

#### OLDSTYLE

Template: OS=OLDSTYLE/S

Tooltype: OLDSTYLE

Use the old (Executive V1.30) style for graphs and frames.

#### MEMORY

Template: MEM=MEMORY/S

Tooltype: MEMORY

Display free memory.

#### CPUUSAGE

Template: CPU=CPUUSAGE/S

Tooltype: CPUUSAGE

Display CPU usage.

#### VMMSTATS

Template: VMM=VMMSTATS/S

Tooltype: VMMSTATS

Display VMM statistics.

#### LIST

Template: LIST/S

Tooltype: LIST

List all keywords.

#### VMEM

Template: VMEM/S

Tooltype: VMEM

If you're are using GigaMem, specify this option. By default the "vmem.library" is not opened, because there are many libraries with the same name, and only the GigaMem's "vmem.library" can be used.

#### NOVOLUMES

---

Template: NV=NOVOLUMES/S  
Tooltype: NOVOLUMES

Don't add volumes to the Diskspace-submenu.

#### NODEVICES

Template: ND=NODEVICES/S  
Tooltype: NODEVICES

Don't add devices to the Diskspace-submenu.

#### LABEL

Template: LB=LABEL/K  
Tooltype: LABEL

The text in the window's titlebar. You can put here whatever you like.

#### NOTITLEBAR

Template: NTB=NOTITLEBAR/S  
Tooltype: NOTITLEBAR

Don't display a titlebar for the window.

#### BORDERLESS

Template: BORD=BORDERLESS/S  
Tooltype: BORDERLESS

No Intuition borders for the window.

#### BACKDROP

Template: BACK=BACKDROP/S  
Tooltype: BACKDROP

Backdrop window, the window is behind all other windows and can't be depth arranged.

#### FONT

Template: F=FONT/K  
Tooltype: FONT  
Default: Screen font

Font to be used in Meter's window. You might want to use a small font, so you can display more information and make the window smaller.

#### FONTSIZE

Template: FS=FONTSIZE/N/K  
Tooltype: FONTSIZE

If you specify a font, you should also specify the font size.

---



## ZIP

Template: ZIP/S  
Tooltype: ZIP

Open Meter's window so that only the titlebar is displayed.

## STAYFRONT

Template: SF=STAYFRONT/S  
Tooltype: STAYFRONT

Keep the window in front of all other windows.

## PUBSCREEN

Template: PS=PUBSCREEN/K  
Tooltype: PUBSCREEN  
Default: default public screen

Name of the public screen where the window is to be opened.

## LEFT

Template: LEFT/N/K  
Tooltype: LEFT

Offset of the left edge of the window relative to screen.

## TOP

Template: TOP/N/K  
Tooltype: TOP

Offset of the top edge of the window relative to screen.

## HELP

Template: HELP/S  
Tooltype: HELP

Display this document.

## VERSION

Template: VERSION/S  
Tooltype: -

Display version number.

## 1.108 Nice

NICE

~~~~~

---

Execute a command with higher nice-value.

OPTIONS  
DESCRIPTION

Nice executes a command with a higher nice-value than the current task has. If no increment is given, 10 is assumed.

Arguments can be given for the command as usual. If you're using a custom shell like csh or sksh, it's possible to run commands under that shell. Several clients use this feature, read about it from [here](#).

There are two ways to give the nice-value increment:

With the NICE-keyword:

```
Nice NICE=10 <command>
```

Or by preceding the value with a '--':

```
Nice -10 <command>      =>  Nice NICE=10 <command>
Nice --10 <command>    =>  Nice NICE=-10 <command>
```

If the server isn't running, Nice will still run the command, but the nice-value is ignored.

Nice can also be used from Workbench, see [Genies](#) for more information.

When MultiUser is installed, only the super-user may use negative nice-value increments. If you don't want to require root privileges for Nice, set the U-bit using MProtect.

## EXAMPLES

Execute dir with nice-value of this task + 10.

```
Nice dir
```

Execute dir with nice-value of this task + (-20).

```
Nice NICE=-20 dir
```

## 1.109 Nice options

NICE OPTIONS

~~~~~

---

## NICE

Template: N=NICE/N/K  
Tooltype: NICE  
Default: 1

Run the command with this task's nice-value plus this increment.

## SHELL

Template: S=SHELL/K  
Tooltype: SHELL  
Default: Run command under standard Amiga shell

If you're using a custom shell, you can specify here a  
template  
that is  
used when starting a command under that shell.

## HELP

Template: HELP/S  
Tooltype: HELP

Display this document.

## VERSION

Template: VERSION/S  
Tooltype: -

Display version number.

## COMMAND

Template: C=COMMAND/F  
Tooltype: COMMAND

The command to be executed and its arguments.

## 1.110 Ps

### PSTREE

~~~~~

Display information about tasks.

### OPTIONS DESCRIPTION

Ps displays information about all or selected tasks in system. Tasks are sorted by task PIDs.

---

Output is controlled with keywords. Each keyword displays a piece of information about a task, for example:

```
PID          task's PID
NAME         task name
TIME         total used cputime
PRI          task's priority
```

And so on, over hundred keywords are currently known. Use the LIST-keyword to get a list of all available keywords.

The default output consists of these keywords:

```
PID,PRI,RPRI,NICE,TYPE,STATE,TIME,IDLE,CPU%,NAME
```

If you would like to display only task's pid, priority and name, you would use this command:

```
Ps FORMAT=PID,PRI,NAME
```

Or just:

```
Ps PID,PRI,NAME
```

See the explanation of  
all available keywords  
.

If you want to align the header of a keyword to left, use < before the keyword and > if you want to align it to right. Some keywords are variable width and you can specify their width with /<width> after the keyword. For example, this will change the width of the NAME-field to 50 characters:

```
Ps PID,PRI,NAME/50
```

These options specify a built-in format string:

```
EVERYTHING   Output every keyword
CPU           Output CPU usage values
STACK         Output stack usage statistics
TASK          Output Exec's task-structure
PROCESS       Output DOS-process structure
CLI           Output CLI-process structure
BABY         Output Executive internal task structure
```

For example,

```
Ps CPU
```

will display several CPU usage values.

You can sort the list by any of these keywords, or even using multiple keywords. For example, if you want to sort by priority, use this command:

```
Ps SORT=PRI
```

If you want that tasks with the same priority are sorted by PID, use this command:

```
Ps SORT=PID,PRI
```

To display the 15 tasks that have been idle for shortest time, tasks must be reverse sorted. This is accomplished with a - in front of the keyword name.

```
Ps SORT=-IDLE NPROCS=15
```

Several options are available for selecting only a part of all tasks for output. For example, the SCHEDULED option displays only scheduled tasks, the WAIT option displays only waiting tasks and the ACTIVE option displays tasks that have recently used CPU time. See the

options

-

section for other similar options.

It's possible to combine keywords like ACTIVE and SCHEDULED, this would display all scheduled tasks that have recently used CPU time:

```
Ps ACTIVE SCHEDULED
```

If you have installed MultiUser and have logged in as some else than root, Ps will display only those tasks that belong to you. To override this, use the option ALL.

CTRL-C will stop the output.

## EXAMPLES

```
Ps
```

Show all devices:

```
Ps NAME=*.device
```

Show stack usage:

```
Ps FORMAT=PID,PRI,NICE,STACK,STACKUSE,STACK%,TIME,CPU%,NAME
```

## 1.111 Ps options

### PS OPTIONS

~~~~~

### FORMAT

Template: F=FORMAT

Tooltype: FORMAT

Default: PID,PRI,RPRI,NICE,TYPE,STATE,TIME,IDLE,CPU%,NAME

The format string. Separate keywords with a comma. Use < before the keyword to align the header to left and > to right. Use /<width> after the keyword to specify width for variable width keywords, e.g. NAME.

This will align the NAME-header to right and the width of this field will be 40 characters:

```
Ps FORMAT=>NAME/40
```

## SORT

```
Template: S=SORT/K
Tooltype: SORT
Default:  PID
```

The sort order. Separate keywords with a comma. Use - before the keyword to sort in reverse with this keyword.

It's sometimes useful to specify multiple keywords here, for example, if you want to sort tasks by CPU%, then tasks with 0 CPU% will be in random order. If you first sort with NAME, these tasks will be sorted by their name:

```
Ps SORT=NAME,CPU%
```

Try this and

```
Ps SORT=CPU%
```

compare the results!

## LIST

```
Template: LIST/S
Tooltype: LIST
```

List all keywords. This will display the keyword name, its header, alignment, width (fixed or variable), default width and minimum width (if variable width keyword).

## NOHEADER

```
Template: NH=NOHEADER/S
Tooltype: NOHEADER
```

Don't display the header.

## PADHEX

```
Template: PH=PADHEX/S
Tooltype: PADHEX
```

Hex numbers will be padded with 0:s, for example, fdc920 becomes 00fdc920.

## PLAIN

---

Template: PLAIN/S  
Tooltype: PLAIN

Don't use ANSI codes in output.

#### DELAY

Template: DELAY/N/K  
Tooltype: DELAY  
Default: 0

Delay specified number of seconds before outputting the task list.  
I added this option so I could see if a screenblanker creates a task  
when it blanks the screen.

#### INVERSE

Template: IV=INVERSE/S  
Tooltype: INVERSE

Inverse task selection. For example:

```
Ps SCHEDULED INVERSE
```

output tasks that are NOT scheduled.

#### STATUS

Template: STATUS/S  
Tooltype: STATUS

Display the same CLI processes as the AmigaOS Status command.

#### ACTIVE

Template: ACTIVE/S  
Tooltype: ACTIVE

Output tasks with non-zero CPU%.

#### IDLE

Template: IDLE/S  
Tooltype: IDLE

Output tasks with zero CPU%.

#### SCHEDULED

Template: SCHED=SCHEDULED/S  
Tooltype: SCHEDULED

Output scheduled tasks.

#### RUN

---

Template: RUN/S  
Tooltype: RUN

Output tasks that are running at the moment. If you don't have a multiprocessor Amiga, this will output only the current task. ;)

#### READY

Template: READY/S  
Tooltype: READY

Output tasks that are ready, but not running.

#### WAIT

Template: WAIT/S  
Tooltype: WAIT

Output tasks that are waiting.

#### ALL

Template: ALL/S  
Tooltype: ALL

If you have installed MultiUser and you have logged in as some else than root, Ps will display only those tasks that belong to you. Using this option you can override this and display all tasks. This option isn't available if MultiUser hasn't been installed.

#### OWN

Template: OWN/S  
Tooltype: OWN

If you have installed MultiUser, this option will display only those tasks that belong to you. This is the default if you have logged in as some else than root. This option isn't available if MultiUser hasn't been installed.

#### PID

Template: P=PID/K  
Tooltype: PID

Display tasks with this PID. You can list multiple PIDs here, separate them with a comma.

#### USER

Template: U=USER/K  
Tooltype: USER

Display tasks that belong to these users, separate user names with a comma. This option isn't available if MultiUser hasn't been installed.

#### GROUP

---



Template: G=GROUP/K  
Tooltype: GROUP

Display tasks that belong to these groups, separate user names with a comma. This option isn't available if MultiUser hasn't been installed.

#### NAME

Template: N=NAME/K  
Tooltype: NAME

Display tasks that match the given name. You can use \* as a wildcard.

#### PGRP

Template: R=PGRP/K  
Tooltype: PGRP

Display tasks in this process group. You can list multiple pgrps here, separate them with a comma.

#### PRIGT

Template: PRIGT/N/K  
Tooltype: PRIGT

Display tasks with greater priority than the given value.

#### PRILT

Template: PRILT/N/K  
Tooltype: PRILT

Display tasks with lower priority than the given value.

#### PRIEQ

Template: PRIEQ/N/K  
Tooltype: PRIEQ

Display tasks with the same priority as the given value.

#### NPROCS

Template: NPROCS/N/K  
Tooltype: NPROCS

Maximum number of tasks to display.

#### EVERYTHING

Template: EVERYTHING/S  
Tooltype: EVERYTHING

This will display information using ALL keywords. You should redirect the output to a file, because it will be hundreds of characters wide.

---

## BABY

Template: BABY/S  
Tooltype: BABY

Display the Executive internal task records, probably not very interesting, I use this for debugging.

## TASK

Template: TASK/S  
Tooltype: TASK

Display the information structure associated with each task.

## PROCESS

Template: PROCESS/S  
Tooltype: PROCESS

Display the information structure associated with each DOS-process.

## CLI

Template: CLI/S  
Tooltype: CLI

Display the information structure associated with each CLI-task.

## CPU

Template: CPU/S  
Tooltype: CPU

Display several CPU usage values of each task.

## STACK

Template: STACK/S  
Tooltype: STACK

Display stack usage values.

## HELP

Template: HELP/S  
Tooltype: HELP

Display this document.

## VERSION

Template: VERSION/S  
Tooltype: -

---

Display version number.

## 1.112 Ps Keywords

### PS KEYWORDS

~~~~~

#### General keywords

##### CPU%

Recent (last minute) CPU usage percentage.

##### CSW

Number of  
context switches  
for this task.

##### CTIME

CPU time used by this task's terminated childtasks.

##### GID

Group ID. Only available if MultiUser has been installed.

##### GROUP

Group name. Only available if MultiUser has been installed.

##### ETIME

Elapsed time since task was created.

##### IDLE

How long task has been idle, without using any CPU time.

##### IVCSW

Number of  
involuntary context switches  
for this task.

##### LSTART

The exact task creation moment, including date.

##### NAME

Task name. Names of CLI processes are in square brackets.

##### NICE

Nice-value for this task.

##### PID

Process ID.

##### PPID

Parent Process ID.

##### PRI

Task's priority.

**QUEUE**

When QUEUES or FEEDBACK scheduler is used this shows task's run queue.

**RPRI**

Task's  
REAL priority  
, not the scheduling priority.

**STACK**

Size of task's stack.

**STACK%**

Percentage of stack used.

**STACKUSE**

How many bytes of stack are currently used.

**START**

The time when task was created.

**STATE**

This will be one of these:

run task is running  
ready task is ready to run  
wait task is waiting

These states are very rare:

invalid something is really wrong with this task  
added task has just been added  
except task is in exception state  
removed task is being removed

**TIME**

Used CPU time.

**TIME%**

Percentage of CPU time used from total available CPU time.  
I.e. if a task has been running for one minute, and it has used 30 seconds of CPU time, TIME% will show 50%.

**TTIME**

CPU time used by this task and its terminated childtasks.

**TYPE**

Task type. This can be:

task task  
proc process  
cli interactive CLI (always a process)  
bcli background CLI (always a process)

**UID**

User ID. Only available if MultiUser has been installed.

**USER**

User name. Only available if MultiUser has been installed.

**VOCSW**

---

Number of  
    voluntary context switches  
    for this task.

#### Task-structure keywords

These keywords display the entire Exec task-structure and aren't probably very interesting if you're not a programmer. They are listed in the same order as they appear in the structure, see <exec/tasks.h>.

#### PRI

Task priority. (tc\_Node.ln\_Pri)

#### TASK

Task-structure address.

#### TNAME

Task name. Not same as NAME, which might be the CLI command name.  
(tc\_Node.ln\_Name)

#### FLAGS

Task's flags: (tc\_Flags)  
  P process  
  T task  
  p proctime  
  e has etask-extension  
  S stackchk  
  x except  
  s switch-vector in use  
  l launch-vector in use

#### STATE

Task state. See above. (tc\_State)

#### NESTING

Interrupt disabled nesting / Task disabled nesting.  
(tc\_IDNestCnt, tc\_TDNestCnt)

#### SIGALLOC

Signals allocated. (tc\_SigAlloc)

#### SIGWAIT

Signals task is waiting for. (tc\_SigWait)

#### SIGRECVD

Signals task has received. (tc\_SigRecvd)

#### SIGEXCEPT

Signals task will take excepts for. (tc\_SigExcept)

#### TRAPALLOC

Traps allocated. (tc\_TrapAlloc)

#### TRAPABLE

Traps enabled. (tc\_TrapAble)

---

EXCEPTDATA  
Points to except data. (tc\_ExceptData)

EXCEPTCODE  
Points to except code. (tc\_ExceptCode)

TRAPDATA  
Points to trap data. (tc\_TrapData)

TRAPCODE  
Points to trap code. (tc\_TrapCode)

STACKPTR  
Stack pointer. (tc\_SPReg)

STACKLOW  
Stack lower bound. (tc\_SPLower)

STACKUPR  
Stack upper bound + 2. (tc\_SPUpper)

SWITCH  
This vector is called when task is losing CPU. (tc\_Switch)

LAUNCH  
This vector is called when task is getting CPU. (tc\_Launch)

USERDATA  
For use by the task. (tc\_UserData)

From V36 upwards Exec has been able to allocate a structure called ETask, which stands for "extended task". The tc\_TrapAlloc and tc\_TrapAble are replaced with tc\_ETask pointer and these two values are moved to the ETask structure. This structure is not currently used, but you can display the structure with these keywords:

ETASK  
ETask address. (tc\_ETask)

ET\_PARENT  
Pointer to task-structure. (tc\_ETask->et\_Parent)

ET\_ID  
ID unique to this task. (tc\_ETask->et\_UniqueID)

ET\_RESULT1  
First result. (tc\_ETask->et\_Result1)

ET\_RESULT2  
Result data pointer (AllocVec). (tc\_ETask->et\_Result2)

#### Process-structure keywords

These keywords display the entire DOS process-structure and aren't probably very interesting if you're not a programmer. They are listed in the same order as they appear in the structure, see <dos/dosextens.h>.

---

SEGLIST  
Array of seg lists used by this process. (pr\_SegList)

PRSTACK  
Process stack size. (pr\_StackSize)

GLOBVEC  
Global vector for this process (BCPL). (pr\_GlobVec)

CLINUM  
CLI task number or zero if not a CLI. (pr\_TaskNum)

PRSTACKUPR  
Pointer to high memory end of process stack. (pr\_StackBase)

RESULT2  
Value of secondary result from last call. (pr\_Result2)

CURDIR  
Lock associated with current directory. (pr\_CurrentDir)

CIS  
Current CLI Input Stream. (pr\_CIS)

COS  
Current CLI Output Stream. (pr\_COS)

CONTASK  
Console handler process for current window. (pr\_ConsoleTask)

FSTASK  
File handler process for current drive. (pr\_FileSystemTask)

CLI  
Pointer to CommandLineInterface. (pr\_CLI)

RETADDR  
Pointer to previous stack frame. (pr\_ReturnAddr)

PKTWAIT  
Function to be called when awaiting message. (pr\_PktWait)

WINDOW  
Window for error printing. (pr\_WindowPtr)

HOMEDIR  
Home directory of executing program. (pr\_HomeDir)

PRFLAGS  
Flags telling dos about process. (pr\_Flags)

- s freeseclist
- d freecurrdir
- c freecli
- i closeinput
- o closeoutput
- f freeargs

---

## EXITCODE

Code to call on exit of program or NULL. (pr\_ExitCode)

## EXITDATA

Passed as an argument to pr\_ExitCode. (pr\_ExitData)

## ARGS

Arguments passed to the process at start. (pr\_Arguments)

## SHELLPRIV

For the use of the current shell. (pr\_ShellPrivate)

## CES

Error stream - if NULL, use pr\_COS. (pr\_CES)

## CLI-structure keywords

These keywords display the entire CLI-structure and aren't probably very interesting if you're not a programmer. They are listed in the same order as they appear in the structure, see <dos/dosextens.h>.

## IOERR

Value of IoErr from last command. (cli\_Result2)

## SETNAME

Name of current directory. (cli\_SetName)

## COMMDIR

Head of the path locklist. (cli\_CommandDir)

## RETCODE

Return code from last command. (cli\_ReturnCode)

## COMMAND

Name of current command. (cli\_CommandName)

## FAIL

Fail level (set by Failat). (cli\_FailLevel)

## PROMPT

Current prompt (set by Prompt). (cli\_Prompt).

## STDINPUT

Default (terminal) CLI input. (cli\_StandardInput)

## CURINPUT

Current CLI input. (cli\_CurrentInput)

## EXECUTE

Name of EXECUTE command file. (cli\_CommandFile)

## CLITYPE

Interactive or background. (cli\_Interactive, cli\_Background)

## CUROUTPUT



Current CLI output. (cli\_CurrentOutput)

DEFSTACK

Stack size to be obtained in long words. (cli\_DefaultStack)

STDOUTOUTPUT

Default (terminal) CLI output. (cli\_StandardOutput)

MODULE

Seglist of currently loaded command. (cli\_Module)

### Executive internal task-structure keywords

These keywords display some parts of the Executive internal task information structure, which is created for each task.

I use these for debugging and they aren't very useful for anybody else, so I don't explain them here.

BABY

BABYNEXT

BABYPREV

BFLAGS

CHILDREN

CPU

CPU2

CSWBAL

CURUTICKS

DADDYUSAGE

MONEY

NEXTCHILD

PARENT

PREVCHILD

RCS

RUSAGE

THASHNEXT

THASHPREV

UTICKS

WATCH

## 1.113 Pstree

PSTREE

~~~~~

Display all tasks in a tree-like structure.

OPTIONS

DESCRIPTION

This client displays the task tree, the child-parent relationships.

You can easily see from the output if a task has created childtasks. Pictures tell more than thousand words, so here is a sample output from Pstree:

```

+---trackdisk.device
+---gvpscsi.device
+---RAM
+---XH0
+---[QMouse]
|   +---[Pstree]
|   |   `---CON
|   |   `---[Make]
|   |       +---CON
|   |       `---[sh]
|   |           `---[ccl]
+---[CyberCron]
`---input.device

```

The root of the tree is at left. Tasks that branch from here don't have a known parent, i.e. they have been created before Executive server was started. To see a more complete tree, you must start the server in s:startup-sequence, right after SetPatch is run. Even then a few tasks are already running.

You can see that QMouse has created two tasks, Pstree and Make. Both are CLI processes, so they have created a CON process. The Make task has also created a shell, which has started a C-compiler.

If you are reading this document with AmigaGuide V39 or newer, you can see that the Pstree and Make tasks are displayed in italics. This means that the task which created them has terminated. In a case like this they are not left orphans, they are transferred to their "grandparents".

With the SHOWPIDS option you can display task PIDs:

```

+---[QMouse] (5)
|   +---[Pstree] (9)
|   |   `---CON (224)
|   |   `---[Make] (8)
|   |       +---CON (236)
|   |       `---[sh] (1)
|   |           `---[cpp] (10)
...

```

Now you can root the tree to a specific PID. This is actually a process group, but in Executive they both mean the same:

Pstree PGRP=8

```

`---[Make] (8)
    +---CON (236)
        `---[sh] (1)
            `---[gcc] (10)

```

## EXAMPLES

Pstree

## 1.114 Pstree options

PSTREE OPTIONS

~~~~~

PGRP

Template: R=PGRP/N  
Tooltype: PGRP

Root the task tree to a specific process group.

SHOWPIDS

Template: SP=SHOWPIDS/S  
Tooltype: SHOWPIDS

Display PIDs in the output.

IBM

Template: IBM/S  
Tooltype: IBM

If you're using IBM style font, try this option, it uses graphical characters from this font to produce nicer output.

PLAIN

Template: PLAIN/S  
Tooltype: PLAIN

Don't use ASCII characters in output.

BACKSLASH

Template: BS=BACKSLASH/S  
Tooltype: BACKSLASH

The ` character may not look too good with all fonts, so with this option you can change it to `.

DELAY

Template: DELAY/N/K  
Tooltype: DELAY

Delay (in seconds) before outputting the tree.

HELP

---

Template: HELP/S  
 Tooltype: HELP

Display this document.

## VERSION

Template: VERSION/S  
 Tooltype: -

Display version number.

## 1.115 Renice

### RENICE

~~~~~

Change task's nice-value.

### OPTIONS DESCRIPTION

Renice is used to increment the nice-value of a task, or multiple tasks at once.

The maximum nice-value is +20, making the task "nice" to others, i.e. it will be given less CPU time than to a task with nice-value -20, which is the minimum value.

I want to stress that the value given to Renice is INCREMENT, not the final value. You can also use negative increments.

Renice can affect multiple tasks at once:

```
Renice 5 PID=10,20 NAME=DH\* USER=root GROUP=staff PGRP=273
```

The following tasks will be affected by this command:  
 (NOTE: Depending on the shell you use, the \ may not be needed in front of the asterisk.)

- tasks with PID 10 and 20
- tasks with name starting with "DH" (case insensitive).
- all tasks owned by the root user. (MultiUser only)
- all tasks owned by the staff group. (MultiUser only)
- all tasks that belong to process group 273.

The new nice-value will be old nice-value + 5.

If you don't specify any task, the nice-value of the current task (usually a shell process) will be affected.

There's no need to use the NICE- and PID-keywords, Renice works without them too:

```
Renice -10 278
```

This will add -10 to the nice-value of the task with PID 278.

The PGRP option is sometimes very useful. Lets say you have a cron-utility installed in your system (I have). Cron is a program that runs in background and executes commands at specific times, as defined by user. I use a cron to regularly take backups of important files and to check for viruses. Pstree shows that I have four cronjobs running:

```
+---[CyberCron] (4)
| +---Virus_Checker(C) (235)
| +---[find] (1)
| +---[LhA] (10)
| `---[tar] (11)
...

```

It's possible to give all these tasks a nice-value +20 with a single command:

```
Renice 20 PGRP=4
```

This will affect the CyberCron task and all its childtasks and also all tasks created by CyberCron from now on.

When MultiUser is installed, only the super-user may use negative nice-value increments. If you don't want to require root privileges for Renice, set the U-bit using MProtect.

## EXAMPLES

Add 10 to the current task's nice-value.

```
Renice 10
```

Force the nice-value to -20 for "RexxMaster". Because the maximum nice-value is +20, we must subtract at least 40 from it to always get to -20. If the nice-value would be 0, this would still be ok, because Renice makes sure that the nice-value doesn't go below -20.

```
Renice -40 name=RexxMaster
```

## 1.116 Renice options

### RENICE OPTIONS

#### NICE

Template: NICE/N

Tooltype: NICE

Increment of nice-value for the specified tasks.

#### PID

Template: P=PID

Tooltype: PID

Default: Current task

PIDs of the tasks whose nice-value is to be incremented. You can list multiple PIDs here, separate them with a comma, for example "PID=10,248,332".

#### NAME

Template: N=NAME/K

Tooltype: NAME

Names of the tasks whose nice-value is to be incremented. You can list multiple names here, separate them with a comma, for example "NAME=DH0,DH1,DH2,\*.device". You can use \* as a wildcard.

#### USER

Template: U=USER/K

Tooltype: USER

Names of the users whose every task's nice-value will be incremented. You can list multiple user names here, separate them with a comma, for example "USER=frodo,pippin,gandalf". You can also use UIDs. This option isn't available if MultiUser hasn't been installed.

#### GROUP

Template: G=GROUP/K

Tooltype: GROUP

Names of the groups whose every task's nice-value will be incremented. You can list multiple group names here, separate them with a comma, for example "NAME=staff,students". You can also use GIDs. This option isn't available if MultiUser hasn't been installed.

#### PGRP

Template: R=PGRP/K

Tooltype: PGRP

PGRPs of the process groups whose nice-value will be incremented. All tasks in the process group will be affected. You can list multiple PGRPs here, separate them with a comma, for example "PGRP=254,298,578".

#### HELP

Template: HELP/S

Tooltype: HELP

Display this document.

VERSION

Template: VERSION/S  
Tooltype: -

Display version number.

## 1.117 Sa

SA

~~~~~

Display system accounting statistics.

OPTIONS  
DESCRIPTION

Sa displays a summary of accounting records and optionally writes this summary to disk. By default, per-command statistics will be displayed. The number of times a command has been executed, total CPU time used and total elapsed time. If you're using MultiUser, you can also get per-user statistics.

Sa maintains two summary files, "acct.procsun" and "acct.usersun". These contain accounting records according to command name and user ID, respectively.

The per-command statistics output looks like this:

COMMAND	CALLS	CPUTIME	CPUTIME%	ELAPSED
NComm30	11	1h50m	23.39%	8h39m
lzx	36	1h09m	14.73%	2h31m
Mosaic	16	19m45s	4.18%	1h49m
locate.sort	6	10m57s	2.32%	23m33s
csch	167	8m48s	1.87%	39h30m
rsc	676	1m08s	0.24%	7m12s
ed	703	41.43s	0.15%	18m40s
newsyslog	6	3.27s	0.01%	29.48s
RX	309	1.17s	0.00%	42.90s
Assign	32	0.96s	0.00%	1.59s
rlog	5	0.91s	0.00%	12.52s
ACUSeeMe_020	1	0.15s	0.00%	3.58s
stack	327	0.00s	0.00%	0.36s
...				

For example, "Mosaic" has been executed 16 times, 19 minutes 45 seconds of CPU time was used, which is 4.18% of all commands in the summary. Mosaic has been in system for 1 hour and 49 minutes.

The per-user statistics output looks like this (if MultiUser has been installed):

USER	COMMANDS	CPUTIME	CPUTIME	ELAPSED
-	8634	7h43m	97.85%	70h34m
root	13	9m14s	1.95%	22m06s
petrin	9	55.48s	0.20%	4m06s

Most of the commands, 8634, were executed without logging in to the MultiUser system, so the username is "-". These commands have used 7 hours 43 minutes of CPU time. The "root" user has used 1.95% of the CPU time and root's commands were in the system for 22 minutes 6 seconds.

It's possible to sort the summary in different ways. The default is to sort the summary by used CPU time, but it's also possible to sort by elapsed time, by command name, by average command execution time or by number of times a command was executed.

When you run Sa it will read the accounting file created by Acct, and the summary files. The new accounting records will be added to the summary. The MERGE option instructs Sa to write the summary back to disk and delete the current accounting file. The summary will be displayed on screen. To see per-user summary use the USER- option.

If you have multiple accounting summary files, you can merge them with Sa, for example:

```
Sa PFILE=acct.procsun.1,acct.procsun.2 UFILE=acct.usersun.1,
  acct.usersun.2 MERGE
```

This will read the "s:acct"-file, "s:acct.procsun", "s:acct.usersun" and the four files named above and merge them to "s:acct.procsun" and "s:acct.usersun". The "s:acct"-file will be deleted. This picture shows what the command does:

```

s:acct.procsun -----.
acct.procsun.1 -----. \
acct.procsun.2 -----. \ \
                        \ \ \
                        .-----./
s:acct ----->| Sa MERGE |-----> display summary on screen
                        \-----/\
                        / / /
s:acct.usersun -----./ / /
acct.usersun.1 -----./ /
acct.usersun.2 -----./

```

If you want to run Sa automatically every day, you need a cron-program, for example CyberCron. You can get it from Aminet, it's in the util/time directory.

To strip garbage (miscellaneous commands executed only a few times) from the summary files, use the JUNK and FORCE options. For example:

```
Sa JUNK=10 FORCE MERGE
```

will move all commands executed 10 or less times to a group "\*\*\*junk" and write the summary files back to disk.



Sa doesn't require that the server is running.

Sa asks the Acct-daemon to write all unpurged records to disk so they will be included in the summary.

CTRL-C will stop the output.

#### EXAMPLES

Show a per-command summary:

Sa

This will join all records to a single group "\*\*\*other":

Sa JOINLONELY THRESHOLD=999999

## 1.118 Sa options

### SA OPTIONS

~~~~~

#### ACCTFILE

Template: F=ACCTFILE  
Tooltype: ACCTFILE  
Default: s:acct

The accounting file name.

#### SUMMARYDIR

Template: SDIR=SUMMARYDIR/K  
Tooltype: SUMMARYDIR  
Default: s:

The directory where the summary files "acct.procsun" and "acct.usersun" are written.

#### AFILE

Template: AFILE/K  
Tooltype: AFILE

This is like the ACCTFILE option, but you can list several accounting files here, separated by a comma. For example: "AFILE=acct.1,acct.2".

#### PFILE

Template: PFILE/K  
Tooltype: PFILE

This is used when merging multiple summary files. You can specify several `procsum-summary` files, separated by a comma. For example: `"PFILE=ap.1,ap.2,ap.3"`.

#### UFILE

Template: UFILE/K  
Tooltype: UFILE

This is used when merging multiple summary files. You can specify several `usersum-summary` files, separated by a comma. For example: `"UFILE=au.1,au.2,au.3"`.

#### MERGE

Template: MERGE/S  
Tooltype: MERGE

Merge the `acct-files` and all given summary files and write the new summary files `"acct.procsum"` and `"acct.usersum"` to `SUMMARYDIR` (default directory for summary files is `s:`).

#### NOACCT

Template: NOACCT/S  
Tooltype: NOACCT

Don't read the accounting-file (`s:acct`).

#### NOSUMMARY

Template: NOSUMMARY/S  
Tooltype: NOSUMMARY

Don't read the summary files. This will output a summary of the accounting-file (`s:acct`).

#### USER

Template: U=USER/S  
Tooltype: USER

Display the by user `-accounting` summary (`acct.usersum`).

#### JOINLONELY

Template: JL=JOINLONELY/S  
Tooltype: JOINLONELY

Join all commands that have been executed only one time to a group called `"***other"`. See the `THRESHOLD` option also.

This only affects the displayed summary.

#### THRESHOLD

Template: TH=THRESHOLD/N/K

---

Tooltype: THRESHOLD  
Default: 1

Used with JOINLONELY option. If command has been executed only as many times as you specify here, it will be joined to a group called "\*\*\*other". For example, "THRESHOLD=10" will join all commands that have been executed 10 times or less to a group "\*\*\*other".

It's not necessary to give the JOINLONELY option with THRESHOLD.

#### QUIET

Template: QUIET/S  
Tooltype: QUIET

Don't display the summary on screen.

#### NOHEADER

Template: NH=NOHEADER/S  
Tooltype: NOHEADER

Don't display the header.

#### SORTBYCALLTIME

Template: SCAT=SORTBYCALLTIME/S  
Tooltype: SORTBYCALLTIME

Sort the summary by average command execution time.

#### SORTBYNAME

Template: SN=SORTBYNAME/S  
Tooltype: SORTBYNAME

Sort the summary by command name.

#### SORTBYCALLS

Template: SC=SORTBYCALLS/S  
Tooltype: SORTBYCALLS

Sort the summary by number of times each command has been executed.

#### SORTBYCTIME

Template: SCT=SORTBYCTIME/S  
Tooltype: SORTBYCTIME

Sort the summary by CPU time.

#### SORTBYETIME

Template: SE=SORTBYETIME/S  
Tooltype: SORTBYETIME

---

Sort the summary by elapsed time.

#### REVERSE

Template: RV=REVERSE/S  
Tooltype: REVERSE

Reverse sort.

#### JUNK

Template: JUNK/N  
Tooltype: JUNK

This option is used to strip garbage from the accounting summary files. The number of times each command has been used will be compared to the value given with this option. For example, JUNK=5 will move all commands that have been executed 5 times or less to a group "\*\*\*junk".

If no FORCE option is given, Sa will print the command name and ask if the command should be "junked". If the reply begins with "y", the command will be moved to the "\*\*\*junk" group.

If you want to write the summary files back to disk, you'll also have to specify the MERGE option.

#### FORCE

Template: FORCE/S  
Tooltype: FORCE

Disable interaction when using the JUNK option. Don't ask if a command should be moved to the "\*\*\*junk" group.

#### HELP

Template: HELP/S  
Tooltype: HELP

Display this document.

#### VERSION

Template: VERSION/S  
Tooltype: -

Display version number.

## 1.119 Timer

### TIMER

~~~~~

Time command execution.

---

OPTIONS  
DESCRIPTION

The Timer-client executes a command and measures its CPU usage. If Timer is started without a command name or it's started from Workbench, it will open a window where you can drop program icons.

Arguments can be given for the timed command. If you're using a custom shell like csh or sksh, it's possible to run commands under that shell. Several clients use this feature, read about it from [here](#)

Timer provides normal or verbose output. The normal output consists of one line of information, it will be displayed after the command has been executed. For example:

```
time 0:00:04.79 + child 0:00:37.83 = total 0:00:42.62, elapsed 0:01:48.98
```

This means that the command consumed 4.79 seconds of CPU time. Its childtasks (i.e. the tasks that it created) used 37.83 seconds of CPU time. This number includes the childtasks that were still running when the main command terminated. The command and its childtasks used 42.62 seconds of CPU time. 1 minute and 48.98 seconds of real time elapsed while the command was running.

The verbose output looks like this:

```
Task time                0:00:05.27 ( 3%)
Childtasks time          0:02:48.79 ( 97%)
Running childtask time   0:00:00.00 ( 0%)
=====
TOTAL USED CPU TIME      0:02:54.06 (100%)

Elapsed time             0:03:55.30
CPU usage                 73.97%
```

This command used 5.27 seconds of CPU time, which is 3% of the total CPU time used by the command and its childtasks. Those childtasks that terminated before the main command used 2 minutes and 48.79 seconds of CPU time, which is 97% of the total used CPU time. No childtasks were running when the main command terminated. 3 minutes 55.30 seconds of real time elapsed while the command was running. Of this time, this command and its childtasks used 73.97% of the total available CPU time.

Timing accuracy is 1/1000 seconds. The CPU time values are very accurate, they don't include the overhead that goes to task creation and loading it from disk. Neither does the elapsed time include any overhead, it's the time that elapses between AddTask() and RemTask() calls.

Timer can be used even if the server isn't running. Then only the elapsed time can be calculated.

When you start Timer from Workbench, or if you have Workbench running and start Timer from a shell without the command name, it will open a window on the Workbench screen. You can drop program icons to this window.

In normal mode, Timer will output the total CPU time used by the command and all its childtasks, elapsed time and CPU usage. You can switch to verbose mode by selecting the menu item Settings->Verbose. The produced output is identical to what is explained above. Window's titlebar displays the number of commands currently being executed.

If the command that you want to run doesn't have an icon, Timer will ask CLI arguments for it.

The Project->Save config menu item will save current settings to the program's icon. This option is available only when the program is started from Workbench.

## EXAMPLES

This will start Timer and open a window on Workbench screen. If you don't have Workbench running, this won't work.

```
Timer
```

This times the c:Dir command:

```
Timer c:Dir
```

This times the c:Dir command and gives verbose output:

```
Timer VERBOSE c:Dir
```

As above, but appends the output to a file T:timer.out

```
Timer VERBOSE OUTPUT=T:timer.out APPEND c:Dir
```

## 1.120 Timer options

### TIMER OPTIONS

#### OUTPUT

```
Template: O=OUTPUT/K
Tooltype: OUTPUT
```

This is used to redirect the output of Timer to a file.  
For example:

```
Timer OUTPUT=T:timer.out dir
```

will output to T:timer.out file.

## APPEND

Template: A=APPEND/S  
Tooltype: APPEND  
Default: overwrite

Used with the OUTPUT option. Instead of overwriting the specified file, append to it.

## VERBOSE

Template: V=VERBOSE/S  
Tooltype: VERBOSE  
Default: output timing information on one line

Tell Timer to give verbose output.

## STDOUT

Template: STDOUT/S  
Tooltype: STDOUT  
Default: output timing information to stderr

By default, timing information is output to standard error stream, which can't be redirected. By using this option, the timing information will be output to standard output stream.

## LEFT

Template: LEFT/N/K  
Tooltype: LEFT

Offset of the left edge of the window relative to screen.

## TOP

Template: TOP/N/K  
Tooltype: TOP

Offset of the top edge of the window relative to screen.

## SHELL

Template: S=SHELL/K  
Tooltype: SHELL  
Default: Run command under standard Amiga shell

If you're using a custom shell, you can specify here a  
template  
that is  
used when starting a command under that shell.

## HELP

Template: HELP/S  
Tooltype: HELP

---

Display this document.

#### VERSION

Template: VERSION/S  
Tooltype: -

Display version number.

#### COMMAND

Template: C=COMMAND/F  
Tooltype: COMMAND

The command to be timed and its arguments.

## 1.121 Top

### TOP

~~~~~

Display and update information about the tasks using most of the CPU time.

#### OPTIONS DESCRIPTION

Top displays the "Top X" tasks using most CPU time. You can specify how many tasks are displayed.

Top will also display system load averages, number of tasks (sleeping, running), CPU usage, current time and available memory.

Top opens a window where it displays this information. The information will be updated periodically (by default every second).

It's also possible to run Top in a shell-window so it can be used from a remote terminal. To display the information once, use the ONCE option. To run Top in a shell-window so that it updates its display, use the STDOUT option. You need a terminal that supports clear screen and cursor positioning commands. If you want to redirect Top's output to a file, use the ONCE option instead of STDOUT. (To see why, try starting Top with the ONCE option and redirect the output to a file. Let it run for a while and break Top with CTRL-C. Now see what the file contains. This might even be useful.)

When Top is running in a shell-window, it reads commands from the terminal and acts upon them accordingly. You need to press Enter after the command. These commands are available:

h or ? Print help text.

---



s or u    Change update interval (seconds).

n         Change number of tasks to display.

i         Toggle the displaying of idle tasks.

d         Redraw the display.

r         Renice tasks. You'll be asked for nice-value increment and one or more PIDs. For example, entering "15 4,5,369" will increment nice-values of tasks 4,5 and 369 by 15. If MultiUser has been installed, then only the super-user may specify a negative nice-value increment.

k         Kill tasks, i.e. send a CTRL-C break signal to them. You'll be asked for one or more PIDs of tasks to kill. With MultiUser, only the super-user may kill other users' tasks.

q         Quit.

Here's an example of Top's display:

```
load averages:  0.63,  0.37,  0.10                               15:51:36
32 tasks: 30 sleeping, 2 running          CPU usage:  18.1%
chip: 758K fast: 5451K virtual: 0K total: 6209K largest: 3966K

  PID USERNAME  PRI NICE  STATE  TIME  CPU NAME
   10 petrin    -89   0 ready  0:06 58.75% [lha]
  216 -         4   0 wait   0:13 18.42% suprascsi.device
  201 -        20   0 wait   0:22  2.44% input.device
    1 root     -70   0 wait   0:07  1.74% [Quickgrab]
    9 -       -51 -15 run    0:01  0.67% [Top]
    6 -       126   0 wait   0:01  0.46% Executive
  202 -        10   0 wait   0:00  0.32% DH2
  211 -         5   0 wait   0:05  0.14% CON
    3 -         5   0 wait   0:01  0.12% [ASwarm]
  223 -        10   0 wait   0:01  0.11% DH0
```

Load averages are similar to what is displayed by  
Uptime

There are 32 tasks in the system, 30 of them are sleeping, 2 are running or ready. The current CPU usage is 18.1%, this is for the last second. The next line displays available memory. Available virtual memory is also displayed (GigaMem or VMM). If you're using GigaMem, use the VMEM option when starting Top.

Task's CPU usage is a weighted average over the last minute. Use the CURRENT option to display current CPU usage instead.

PID is process identifier. USERNAME is displayed if MultiUser has been installed. PRI is task priority and NICE is task's nice-value.

STATE is wait, ready or run. TIME is the amount of CPU time the task has used. NAME is the task name. If it's in square brackets, it's a CLI command name.

The menu item Project->New will redraw the display. The menu item Settings->Update sets the update interval.

To display current CPU usage instead of recent CPU usage, choose Settings->Current CPU usage.

If you set the Settings->No idle tasks menu item, idle tasks won't be displayed. The Settings->Stay front keeps Top's window in front of all other windows.

When you double-click a task, Top asks you if you want to send a break signal to that task. If MultiUser has been installed, then only super-user may break other users' tasks.

When you press the zoom gadget, the window's height is reduced so that only the header is displayed.

The Project->Save config menu item will save current settings to the program's icon. This option is available only when the program is started from Workbench.

The window has invisible drag, close and depth gadgets if the system gadgets have been disabled with NOTITLEBAR option. The centre of the window works as a drag-bar. The SIZEGADGET option adds a size gadget to the window.

Top, when running on a separate window, is a commodity and supports

screennotify.library

.

## EXAMPLES

Top

Like above, but show 15 tasks:

Top 15

With size gadget:

Top SIZEGADGET

Show as many tasks as possible:

Top MAX

Instead of recent CPU usage, display current CPU usage:

Top CURRENT

Just output the information and then exit:

---

Top ONCE

Run Top in the current shell-window, press CTRL-C or 'q' + ENTER to quit:

Top STDOUT

Don't show idle tasks, update every 5 seconds:

Top NOIDLE UPDATE=5

Use a different font:

Top FONT=courier FONTSIZE=13

## 1.122 Top options

### TOP OPTIONS

#### TASKS

Template: T=TASKS/N  
Tooltype: TASKS  
Default: 10

Number of tasks to display.

#### MAX

Template: MAX/S  
Tooltype: MAX

Show as many tasks as possible.

#### CURRENT

Template: CUR=CURRENT/S  
Tooltype: CURRENT

Display current CPU usage instead of recent CPU usage.

#### NOIDLE

Template: NI=NOIDLE/S  
Tooltype: NOIDLE  
Default: Show idle tasks also

Don't show idle tasks.

#### UPDATE

Template: UD=UPDATE/N/K

---

Tooltype: UPDATE  
Default: 1

The update interval (in seconds).

#### FONT

Template: F=FONT/K  
Tooltype: FONT  
Default: System default font

Font to be used in Top's window. You might want to use a small font, so you can display more tasks. Top can only use mono-spaced, non-proportional fonts.

#### FONTSIZE

Template: FS=FONTSIZE/N/K  
Tooltype: FONTSIZE

If you specify a font, you should also specify the font size.

#### BGCOLOR

Template: BGC=BGCOLOR/N/K  
Tooltype: BGOLOR

Background color. (0-255)

#### TEXTCOLOR

Template: TXTC=TEXTCOLOR/N/K  
Tooltype: TEXTCOLOR

Text color. (0-255)

#### ZIP

Template: ZIP/S  
Tooltype: ZIP

Open Top's window so that only the header is displayed.

#### SIZEGADGET

Template: SG=SIZEGADGET/S  
Tooltype: SIZEGADGET

Add size gadget to the window. When used together with NOTITLEBAR, the size gadget will be invisible.

#### STDOUT

Template: S=STDOUT/S  
Tooltype: STDOUT

Don't open a window, display the information in the current shell-

---

window.

#### BATCH

Template: BATCH/S  
Tooltype: BATCH

Disable reading keyboard when information is displayed in a terminal.

#### ONCE

Template: ONCE/S  
Tooltype: ONCE

Don't open a window, just output the information and exit.

#### UIDS

Template: UIDS/S  
Tooltype: UIDS

Display UIDs instead of user names.

#### VMEM

Template: VMEM/S  
Tooltype: VMEM

If you're are using GigaMem, specify this option. By default the "vmem.library" is not opened, because there are many libraries with the same name, and only the GigaMem's "vmem.library" can be used.

#### NOTITLEBAR

Template: NTB=NOTITLEBAR/S  
Tooltype: NOTITLEBAR

Don't display a titlebar for the window. Instead, use invisible close, depth, drag and size (if SIZEGADGET option is specified) gadgets.

#### CX\_PRIORITY

Template: CX\_PRIORITY/N/K  
Tooltype: CX\_PRIORITY  
Default: 0

Commodity priority (from -128 to 127).

#### CX\_POPUP

Template: CX\_POPUP/K  
Tooltype: CX\_POPUP  
Default: yes

This can be "yes" or "no". If it's "no", then Top will not open its window when it is started.

---

## CX\_POPKEY

Template: CX\_POPKEY/K  
Tooltype: CX\_POPKEY  
Default: no hotkey

The hotkey which will open Top's window closed with the Project->Hide command. For example, "ctrl alt t".

## BACKDROP

Template: BACK=BACKDROP/S  
Tooltype: BACKDROP

Backdrop window, the window is behind all other windows and can't be depth arranged.

## NOSCREENNOTIFY

Template: NOSCREENNOTIFY/S  
Tooltype: NOSCREENNOTIFY

Don't use  
screennotify.library  
.

## STAYFRONT

Template: SF=STAYFRONT/S  
Tooltype: STAYFRONT

Keep the window in front of all other windows.

## PUBSCREEN

Template: PS=PUBSCREEN/K  
Tooltype: PUBSCREEN  
Default: default public screen

Name of the public screen where the window is to be opened.

## LEFT

Template: LEFT/N/K  
Tooltype: LEFT

Offset of the left edge of the window relative to screen.

## TOP

Template: TOP/N/K  
Tooltype: TOP

Offset of the Top edge of the window relative to screen.

## HELP

Template: HELP/S  
 Tooltype: HELP

Display this document.

#### VERSION

Template: VERSION/S  
 Tooltype: -

Display version number.

## 1.123 Stat

### STAT

~~~~~

Display miscellaneous statistics.

### OPTIONS DESCRIPTION

This is a sample output from the Stat-client:

Executive 1.20

Registered to: Petri Nordlund [#1]

Boottime: Tuesday 11-Apr-95 14:44:59  
 Uptime: 27 mins  
 Load average: 0.41 0.69 0.48  
 Total used CPU time: 0:15:49.55 (62.88%)  
 Total idle CPU time: 0:09:20.45 (37.12%)  
 Context switches: 243841

Tasks created/finished: 7/3  
 Tasks running/sleeping: 2/27  
 Number of clients/handlers: 1/0  
 Number of known windows: 2

Scheduler:	Feedback	Timer:	CIA-B
Catch range:	-105/1	Dynamic range:	-70/-50
Time resolution:	1/1000s	Watched tasks:	6
Scheduling:	Enabled		

Seconds before purge:	100	Max records in memory:	8
Unpurged records:	0	Accounting daemon pid:	7
Records created:	247		

Stat outputs miscellaneous information about the system and Executive:

Registered to

If your name isn't here, then shame on you! ;-)

#### Bovertime

The time when your system was booted.

#### Uptime

How long your system has been up.

#### Load average

System load averages for 1, 5 and 15 minutes.

#### Total used CPU time

How much CPU time has been used and how many percentages that is from uptime.

#### Total idle CPU time

How long time the CPU has been idle.

#### Context switches

Number of context switches been made.

#### Tasks created/terminated

Number of tasks added and removed from the system.

#### Tasks running/sleeping

Number of tasks currently running or sleeping.

#### Number of clients/handlers

Number of clients currently connected to the server. A handler is created by a client that needs to update its display frequently (e.g. ALoad and Top).

#### Number of known windows

How many opened windows have a known owner.

#### Scheduler

Currently used  
scheduler  
.

#### Timer

Currently used  
timer  
.

#### Catch range & Dynamic range

See the  
server documentation  
Time resolution  
The accuracy of CPU usage measurements.

#### Watched tasks

How many task's are currently being  
watched  
.

#### Scheduling

Scheduling can be  
disabled  
using a commodities exchange program.

#### Accounting is deactivated

Tells you if accounting is off.

#### Seconds before purge

How many seconds to wait (maximum) before writing records to disk.

#### Max records in memory

Maximum number of records held in memory at once.

#### Unpurged records

Records currently in memory

---



Accounting daemon pid  
 PID of the Acct daemon.  
 Records created  
 Total number of accounting records created.

EXAMPLES

Stat

### 1.124 Stat options

STAT OPTIONS

HELP

Template: HELP/S  
 Tooltype: HELP

Display this document.

VERSION

Template: VERSION/S  
 Tooltype: -

Display version number.

### 1.125 Uptime

UPTIME

Display current time, length of time the system has been up,  
 number of users, and system load averages.

OPTIONS  
 DESCRIPTION

Uptime displays current time, length of time the system has been up, number of users (with MultiUser), and load averages over the last 1, 5, and 15 minutes on a single line. If started from Workbench, a window will be opened and the information will be displayed in window's titlebar. This information will be updated by default every second.

Unlike other Uptime utilities for Amiga, this one displays correct Uptime, there's no need to run any programs or daemons when the system

is booted.

Uptime is displayed as "up 10 mins", "up 1:43" (one hour, 43 minutes) or "up 7 days".

The number of users currently logged in is got from multiuser.library. It's only shown if MultiUser has been installed.

The system load averages are displayed as three values, for example:

```
load: 2.12, 0.67, 0.15
```

This means that over the last minute, approximately 2.12 tasks have been running or ready to run. The higher the number, the higher the load of your computer is, i.e. the more tasks there are fighting for CPU time. The other two values are for 5 and 15 minutes, respectively.

When Uptime is run from Workbench, it opens a window and displays this information in the window's titlebar. You can choose the update interval from menus, as well as what information should be displayed.

The Settings->Stay front menu item, when set, keeps the Uptime window in front of all other windows.

The Project->Save config menu item will save current settings to the program's icon. This option is available only when the program is started from Workbench.

```
Uptime is a commodity and supports
    screennotify.library
    .
```

## EXAMPLES

```
Uptime
```

```
Uptime NOTIME NOUSERS
```

```
Uptime WINDOW
```

```
Uptime WINDOW NOTIME NOUSERS NOUPTIME LEFT=50
```

## 1.126 Uptime options

### UPTIME OPTIONS

#### WINDOW

```
Template: W=WINDOW/S
Tooltype: NOWINDOW
Default:  Use window if started from Workbench
```

If Uptime is started from Workbench, it will open a window to show the information. This can be overridden with the NOWINDOW-tooltype. If you start Uptime from a shell and want it to use a window, use the WINDOW option.

#### NOTIME

Template: NOTIME/S  
Tooltype: NOTIME

Don't display the current time.

#### NOUPTIME

Template: NOUPTIME/S  
Tooltype: NOUPTIME

Don't display Uptime.

#### NOUSERS

Template: NOUSERS/S  
Tooltype: NOUSERS

Don't display the number of users currently logged in.

#### NOLOAD

Template: NOLOAD/S  
Tooltype: NOLOAD

Don't display system load averages.

#### UPDATE

Template: UD=UPDATE/N/K  
Tooltype: UPDATE  
Default: 1

Update interval in seconds, used only when the information is displayed in a window.

#### CX\_PRIORITY

Template: CX\_PRIORITY/N/K  
Tooltype: CX\_PRIORITY  
Default: 0

Commodity priority (from -128 to 127).

#### CX\_POPUP

Template: CX\_POPUP/K  
Tooltype: CX\_POPUP  
Default: yes

This can be "yes" or "no". If it's "no", then Uptime will not open its window when it is started.

#### CX\_POPKEY

Template: CX\_POPKEY/K  
Tooltype: CX\_POPKEY  
Default: no hotkey

The hotkey which will open Uptime's window closed with the Project->Hide command. For example, "ctrl alt u".

#### NOSCREENNOTIFY

Template: NOSCREENNOTIFY/S  
Tooltype: NOSCREENNOTIFY

Don't use  
screennotify.library  
.

#### STAYFRONT

Template: SF=STAYFRONT/S  
Tooltype: STAYFRONT

Keep the window in front of all other windows.

#### LEFT

Template: LEFT/N/K  
Tooltype: LEFT

Offset of the left edge of the window relative to screen.

#### TOP

Template: TOP/N/K  
Tooltype: TOP

Offset of the top edge of the window relative to screen.

#### PUBSCREEN

Template: PS=PUBSCREEN/K  
Tooltype: PUBSCREEN  
Default: default public screen

Name of the public screen where the window is to be opened.

#### HELP

Template: HELP/S  
Tooltype: HELP

Display this document.

---

VERSION

Template: VERSION/S  
Tooltype: -

Display version number.

## 1.127 About custom shells

ABOUT CUSTOM SHELLS

~~~~~

These two clients execute shell commands:

- \* Nice
- \* Timer

Commands are normally run under the standard Amiga shell, but if you're using a custom shell, you can run commands under it, and even run that shell's internal commands. This is accomplished with an environmental variable called EXECUTIVESHELL.

This variable contains a shell command which is used to start the command under a custom shell. Here are some examples:

```
csh:    csh -c %c
sksh:   sksh -c %c
ksh:    ksh -c %c
```

So, if you're using the csh-shell, set the EXECUTIVESHELL to "csh -c %c" with these commands:

```
setenv EXECUTIVESHELL "csh -c %c"
copy ENV:EXECUTIVESHELL ENVARC:
```

The %c is substituted with the command name and its arguments.

You can now run internal shell commands and use shell's aliases and even pipes, if they're supported by the shell. You may have to use \| instead of | if you want to use pipes this way because the command line is parsed twice.

If you need to use the % character in the command, use %% instead.

I'm using csh and I have this command in my s:.cshrc:

```
cd A:Com/D1
```

This will change the directory to A:Com/D1 so that when I start a new shell, it will always be in the same directory. I have to use this EXECUTIVESHELL variable:

```
csh -c "cd %d ; %c"
```

The %d is substituted with the current directory. This is needed because otherwise csh would try to start all commands from A:Com/Dl-directory, not from the directory I have run Timer or Nice client. For example, if I'm in directory WORK:Programs and do this:

```
WORK:Programs> Timer lha x TMP:archive.lha
```

Timer will execute this command:

```
csh -c "cd WORK:Programs ; lha x TMP:archive.lha"
```

This ensures that the archive is extracted to the directory WORK:Programs.

If you're using some other shell that isn't mentioned above, tell me what EXECUTIVESHELL variable you are using so I can add it here.

---