The Complete Amiga 500 User Guide

© Copyright Marius Lauritzen & Trond Pedersen Productions 2001

The Complete Amiga 500 User Guide

by Peter Hutchison (2003)

Introduction

Welcome to the Commodore Amiga A500, one of the most popular Amiga models of its time. It was affordable, easy to use and good software in particular games which Jay Minor, the creator of the Amiga, had designed it for.

The Amiga A500 is based on the Motorola 68000 7MHz Processor with 512K (or more) RAM, a single 880K floppy drive with support for 3 more floppy drives, a Custom Chipset that provides the Sound and Graphics.

The Amiga runs the Operating System called AmigaOS which consists of the Kickstart ROM which contains some essential libraries and devices needed to load Workbench which is the desktop:

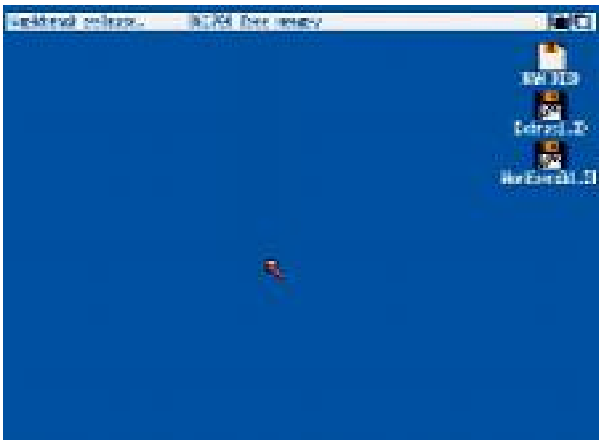


Figure 1

You can a while menu bar at the top and all the disks mounted on the right hand of the screen. The Ram Disk is a special one which is a disk in memory basically. More on Workbench will be explained later.

Setting up the Amiga for First Time

Before setting up the Amiga make sure you have the following items ready:

Amiga A500
Monitor or TV Set
Mouse
Power Supply
Joystick (optional)
Speakers (optional)
External Disk drives (optional)

Withe Amiga facing you, first plug the square end of the Power Supply cable to the Power socket on the top left hand at the back of the Amiga. Plug the other end into a power socket but DO NOT switch on yet.

If you have a monitor plug the monitor cable into the Video socket, if you have a TV Set plug it into the RF or Composite socket and also plug the power cable into power socket.

Next plug the Mouse into Port 1 of the D shaped sockets on the right hand side a the back. Plug any Joystick onto Port 2 next to it.

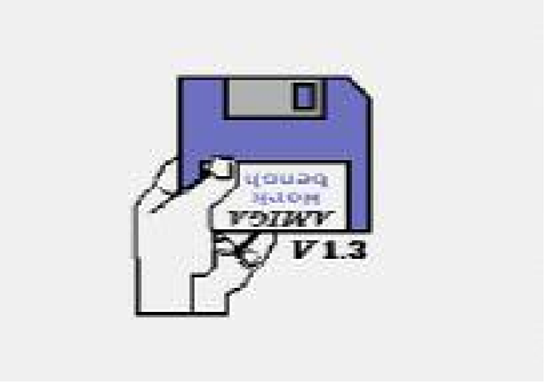
The stereo speakers can be plugged into the Left and Right speaker sockets in the middle at the back. The speakers can either be stand alone ones or part of the monitor so if necessary plug the other ends into the Left/Right sockets on the monitor. Plug in to power supply as necessary.

Finally if you have any external floppy disk drives plug them in to the External Drive socket on the back of the Amiga. Further drives can be daisy chained to other drives. If you have an external hard disk, remove the cover from the Expansion port on the left hand side of the Amiga and slot in the drive.

Now switch on the power and press the power switch located on the Power Supply for the Amiga and switch on the monitor.

If you do not have disks inserted you will see the Insert Disk screen below:

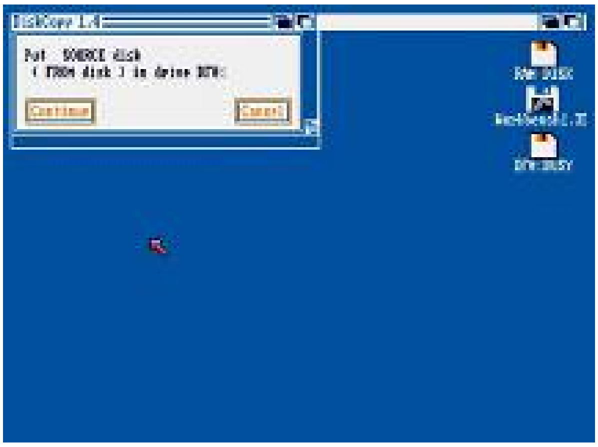
Figure 2



You can then either insert a Workbench disk or any other bootable disk such as a game into the floppy drive on the right hand side of the Amiga (called DF0: - Disk Floppy Zero).

Insert your Original Workbench disk and wait for the workbench screen to appear. Before continuing, I recommend that you make one or more copies of your original Workbench disks to use rather than the Original as that can become damaged over time and you may need to make other copies later.

To make a copy, get hold of a blank 880K double-density disk ready, then move the white arrow (called a pointer) with the mouse over the Workbench 1.3 disk and click the left mouse button once (this will select the disk), then move the pointer to the top left of the screen and click and HOLD the right hand mouse button and a menu appear, make sure the Workbench menu is selected, move the mouse down the menu and select Duplicate



Put SOURCE disk
(FROM disk) in drive DF0:

Figure 3

It will say ‘Put the SOURCE disk (FROM disk) in drive DF0:’, remove the disk from the floppy drive, and make sure it is your original Workbench disk, at the top there is a hole, make sure it is covered with the black tab to protect the disk during this operation. Reinsert the disk, and with the left mouse button click once on Continue.

Once the read operation is complete, it will ask you to insert the DESTINATION (TO disk) in drive DF0:. Press the eject button to remove your original Workbench disk, and insert your Blank disk. Click on Continue to write the information to your new disk.

You may need to repeat the disk swap a couple of times more until the opertation is completed. Now you will have a disk called ‘copy of Workbench’, you can rename this by selecting the disk, then select Rename from the Workbench menu and change it to just ‘Workbench’. You should put your original disk somewhere safe and then use your copy of Workbench from now on. If necessary, reboot with the new Workbench disk in DF0.

Guide to Workbench 1.3

Contents

1. The Menus

At the top of the screen, there are three menus which are accessible when you move the pointer to the top of the screen and click the Right Mouse Button. (RMB).

1.1 Workbench menu

1.1.1 Open

This is used on a selected icon. Instead of double clicking on an icon to show contents of a disk or drawer or run a program you can select an icon with the pointer and Select Open to display or run it.

1.1.2 Close

Instead of clicking on the Close Gadget at the top left of a disk or drawer window just select the window (borders blue) and select Close.

1.1.3 Duplicate

This open allows you to copy files and disks. To copy a file select the icon and then select Duplicate. Another icon will appear saying Copy_of_file or whatever, you can then move that file elsewhere or Rename it.

1.1.4 Rename

This option allows you to change the name of a disk, a drawer or a file. Select the disk, drawer or file icon to rename, select rename, and then type in the new name in the given requester.

1.1.5 Info

This option displays details of a given disk, drawer or file. Workbench has 5 types of workbench icons: Disk, Drawer, Trashcan, Project and Tool. A Trashcan is a special drawer for use with the 'Empty Trash' feature to delete files. A Project is a data file and a Tool is a Program. Info will display the type, size, stack size, default file, comment and protection flags of a file.

1.1.6 Discard

This is a delete function, to delete a file you would normally drag and drop the file into the Trashcan drawer to empty later. This option allows you to delete a file straight away. A requester may appear to confirm deletion.

1.2 Disk Menu

1.2.1 Empty Trash

This option deletes all the files in a selected Trashcan drawer of a disk. This feature is useful because you may want to delete a file but you could change your mind and decide to keep it, all you have to do is to drag the file out of the Trashcan back to its original drawer or disk.

1.2.2 Initialise

This option formats a disk for use by the Amiga. It splits the disk into the right number of tracks/sectors and creates an empty root directory. It can also automatically create a Trashcan drawer. To use, select the disk to format and select Initialise.

1.3 Special Menu

1.3.1 Cleanup

This option rearranges icons on a drawer or disk into a less cluttered state. To use, open the window to rearrange and select Cleanup. To keep the icons in the new positions, select all the icons and select 'Snapshot'.

1.3.2 Last Error

This will display the last message that appeared in the menu bar at the top of the screen.

1.3.3 Redraw

If the screen or window has been corrupted by another program and not all the icons or windows have been drawn then this option will do it for you.

1.3.4 Snapshot

This option allows you to save the position of an icon on a disk or a drawer. To use, select the icon, drag the icon to its new location and Snapshot it.

1.3.5 Version

This will display the version of Workbench and Kickstart you are using. Commodore uses release numbers. For 1.3 it is release 34.

2. The Mouse

On the Amiga mouse there are two buttons. The Left Mouse Button (LMB) and the Right Mouse Button (RMB). Moving the mouse will move the arrow or pointer on the screen.

2.1 Using the Right Mouse Button

To use the menus, press down the RMB and move the pointer to the title bar at the top of the screen. Move the pointer over one of the menu names and some menu items will appear. Now, still with the RMB pressed, move the pointer down to the selected menu item and release the RMB to select that item.

2.2 Using the Left Mouse Button

This button is mainly used to select and move icons around and click on gadgets (or buttons).

2.2.1 Clicking

To select an icon move the pointer over the desired icon and then click the LMB once. The icon will change to a reverse or an alternative image to indicate it has been selected.

When using gadgets such as those on windows (there are four gadgets on a window: Close, Window-to-Front, Window-to-Back and Resize) move the pointer over the desired gadget and click once to use that gadget. The gadget will change briefly and also change the window.

2.2.2 Double-Clicking

This is sometimes a tricky technique to use and requires a steady hand. This method is used to open disks, drawers and files or run programs. To do this, move the pointer over the desired icon and in quick succession (without moving the pointer) click on the LMB twice.
This should open the icon.

2.2.3 Dragging

This technique allows you to move objects such as icons around the screen or resizing windows. Select an icon with a single click, now keep the LMB down and move the pointer, the icon will now be dragged with the pointer. If you let go of the pointer the icon will be dropped into its new position.

2.2.4 Multiple selection

You can select more than one icon at a time. To do this, click on the first icon, press the SHIFT key down (either one) and then click on the second, third, fourth icons etc to select multiple icons

You can also select a whole group of icons by pressing the LMB at the top left of the icons and while keeping the LMB down moving the power towards the bottom right. A expanding box will apear and the icons in it will be selected.

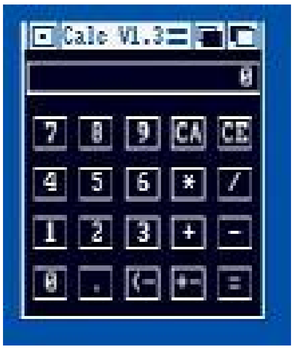
3. WB Programs

Here I will give a brief explanation of the programs provided with Workbench 1.3.

3.1 Workbench Disk

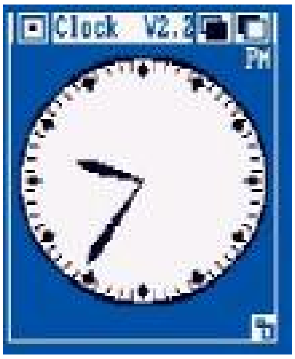
3.1.1 Utilities Drawer

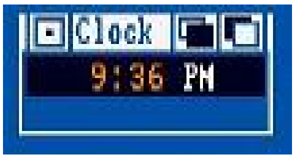
3.1.1.1 Calculator

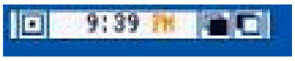


Here is a simple calculator, you can press the buttons to enter values or use the keypad. The symbols are CA (Cancel All), CE (Cancel Entry), * (Multiply), / (Divide), + (Add), - (Subtract), . (Decimal point), <- (Backspace), +- (Change sign), = (Equals)

3.1.1.2 Clock

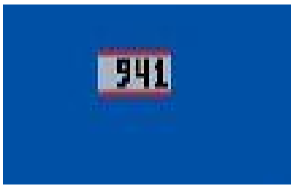






Displays an analogue or digital clock showing the present time and/or date. From the menu you can select Analog, Digital 1 or Digital 2 (difference being is that one displays time in a window, the other displays time in window bar). You can also set the mode (12 or 24 hour), display seconds or not, the date and whether to set an alarm for something.

3.1.1.3 ClockPtr



Turns the pointer from an arrow to a clock! Run Clockptr again to turn it off.

3.1.1.4 Cmd

Redirects output to the printer to a file. By default the file is RAM:CMD_File and redirect anything printed to parallel port to the file. To change the options, select the Cmd icon, select Workbench, Info and then change the ToolTypes. Useful if you want to print something out but have no printer, then you can copy file to another machine, and copy the file to PRT: or PAR: port for printing.

Usage: Cmd device filename [opt i s m n]

3.1.1.5 Graphic Dump

Prints out graphics and pictures. To use this tool, you can either run it from a command Shell and enter filename after Graphic Dump or select a Picture and then Ctrl or Shift double click the GraphicDump icon to print the picture. It must be a IFF/ILBM picture for it to work.

Usage: GraphicDump [Tiny | small | medium | large | xdots:ydots]

3.1.1.6 Install Printer

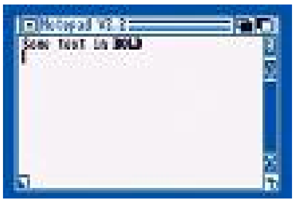
Installs Printer driver in devs/printers from the Extras1.3 disk. Run the program, enter name of printer to install and click Enter or press just Enter to exit.

3.1.1.7 More

Program to display text files. You can run this from Workbench or command Shell or just use it as the default tool for text files so when you double click them, it will display the file for you. If you run it from Workbench it will ask you ‘Filename?’ and you need to type full path to file. You can also select your file, and then Shift/Double click More to display the file.

Usage: More filename or More <PIPE:pipename

3.1.1.8 Notepad



Some text in BOLD

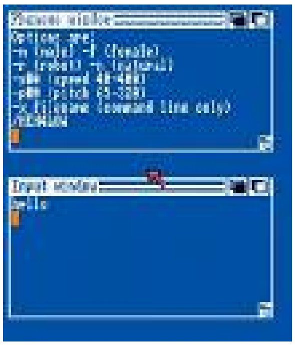
A simple text editor which you can enter short pieces of text and save them to disk, print them out and so on. Unlike other text editors, you can also do Find and Replace, select different Fonts, Colors and Styles to use for your text from the menus.

3.1.1.9 Printfiles

Prints text files to the printer. You can either use the command Shell or from Workbench by selecting your text file and Shift/Double clicking on the Printfiles icon.

Usage: Printfiles [-f] file [-f] file ....

3.1.1.10 Say



```
Phoneme window
Options are:
-m (male) -f (female)
-r (robot) -n (natural)
-s## (speed 40-400)
-p## (pitch 65-320)
-x filename (command line only)
[illegible]
```

```
Input window
hello
```

Speaks any text you type in the Input Window and displays and speaks the Phoneme version in the second window. Options include: -m | -f (male or female voice), -n | -r (natural or robot voice), -s to set speed of speech (40-400) and the pitch (-p) from 65 to 320.

3.1.2 System Drawer

3.1.2.1 CLI

This opens a Command Line Interface console where you can type commands instead of using Workbench. You can launch Workbench programs as well. The location of common commands is kept in the Path (which usually consists of C:, SYS:Utilities, SYS:System and Current Directory), if the command is not found then an error is produced. To exit the Shell, type ENDCLI.
A list of commands is given later in this guide.

3.1.2.2 Diskcopy

Program to copy one disk to another disk. In Workbench, you can select a disk icon and select Duplicate which will use Diskcopy to make an exact copy of the disk, or drag a disk onto another disc to duplicate it. If using the command shell, then you can copy disks using the following command:

Usage: Diskcopy [FROM] <disk> [TO] <disk> [NOVERIFY] [MULTI] [NAME <name>]

where disk is name of drive to copy to/from, Noverify will miss the verify copy stage, Multi will allow you to make multiple copies from same source and Name will allow to specify a new name for new disk.

3.1.2.3 Format

Formats a blank disk with the Amiga Old FileSystem or FastFileSystem to be used to save files onto. AmigaOS can use double sided floppy disks can format them to 840K or 880K (FFS) with the Format command. When you use Workbench to format a disk, you can use the Initialise menu option.
Format can be run from the command shell:

Usage: Format DRIVE <disk> NAME <name> [FFS] [NOICONS] [QUICK]

You can specify the name of the drive e.g DF0:, the name of the new disk and whether to use the Fast File System (FFS), not to create a Trashcan icon (NOICONS) and where to erase a disk using QUICK method.

3.1.2.4 Fastmemfirst

Running this program will tell AmigaOS to load programs into Fast memory first instead of the slower Chip RAM. You can insert this into your startup-sequence if necessary if you have any Fast memory installed. To see if you have any Fast memory installed use the AVAIL command.

3.1.2.5 SetMap

Selects the keyboard mapping to map symbols to keys on your keyboard. To set a map, make sure the map file is in devs/keymaps and add a ToolType in the icon, for example, KEYMAP=gb. Also, you can use it from the command shell:

Usage: Setmap <map>

3.1.2.6 InitPrinter

The command sends an initialiasation string to the printer, similar to switching printer off and on.
.

3.1.2.7 NoFastMem

This command disables all fast memory which may be useful for old programs that will not work with fast memory installed. Re-run program to re-enable fast memory.

3.1.2.8 Mergemem

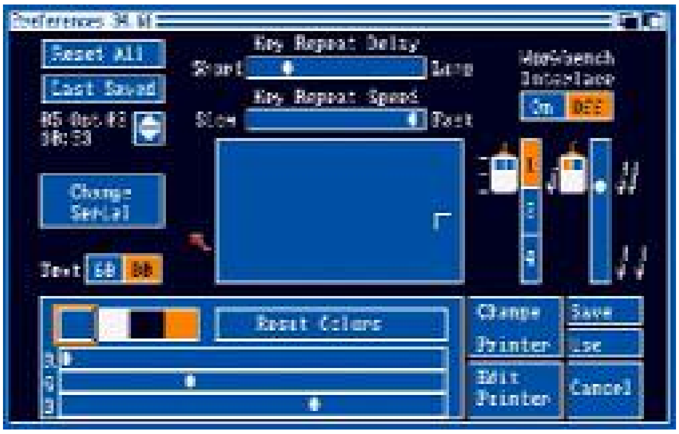
This program adds other memory to be used by the Amiga which does not automatically configure it self to be added to the Amiga’s memory pool.

3.1.2.9 Fixfonts

If you add new fonts or font sizes then this program will update Fonts files if changes have been made to any files in Fonts drawer.

3.1.3 Prefs Drawer

3.1.3.1 Preferences



Changes the settings used by the Amiga such as the pointer, printer, serial port etc.

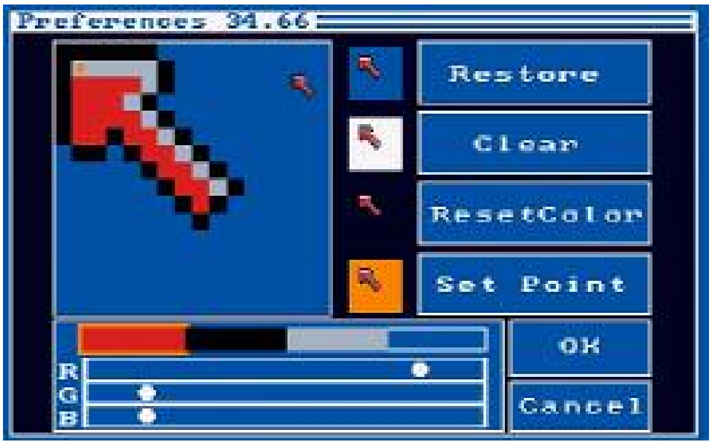
ResetAll	- Resets all settings back to the default
Last Saved	- Resets all settings back to last saved settings
<date time>	- Sets the current date and time. For it to be updated, the Amiga needs to have a battery backed up clock which it does not have by default.
Change Serial	- Change Serial Port settings (see below)
Text 60:80	- Change Text to 60 or 80 columns per line
Key Repeat delay	- Sets how long to wait before a key repeats if held down
Key Repeat speed	- Sets how fast a key keeps repeating
Centre Screen	- Moving the small box in large box will move the screen position

Workbench Interlace - Turns screen interlace mode on or off.
Mouse 1:2:4 - Sets mouse movement speed. 1 is fastest, 4 is slowest
Mouse Click - Sets mouse double click speed
Reset Colors - Resets colours back to defaults
RGB - Changes colour for the four Workbench colour palette (default is Blue, White, Black and Orange)
Change Printer - Change Printer settings (see below)
Edit Pointer - Change the Pointer shape and settings (see below)
Save - Save new settings (this will save them in devs:system-configuration)
Use - Sets settings but does not save them to disk (useful for temporary settings or for trying them out)
Cancel - Cancel preferences without keeping any changes

3.1.3.2 CopyPrefs

Makes a copy of the preference file (system-configuration) another disk in the Devs drawer. Useful for backups to another Workbench disk.

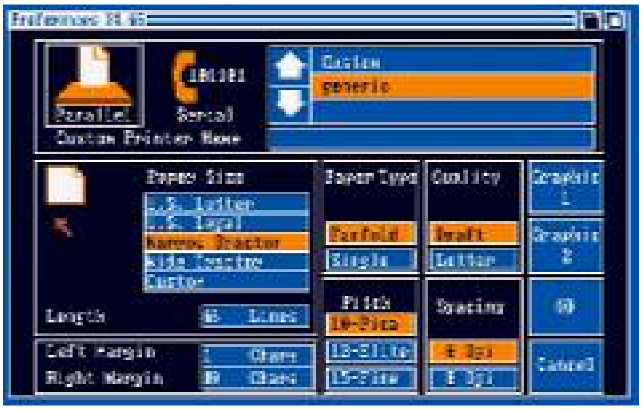
3.1.3.3 Pointer



Displays the pointer preferences screen.
You can modify the pointer's shape in the top left section of the screen, and change the colours if necessary.

Restore - Restores pointer back to default shape
Clear - Clears the drawing area completely
ResetColor - Resets colours back to the defaults
Set Point - Sets the click point (indicated by a small orange square at tip of pointer)
OK - Sets and saves pointer changes
Cancel - Cancel changes and exits

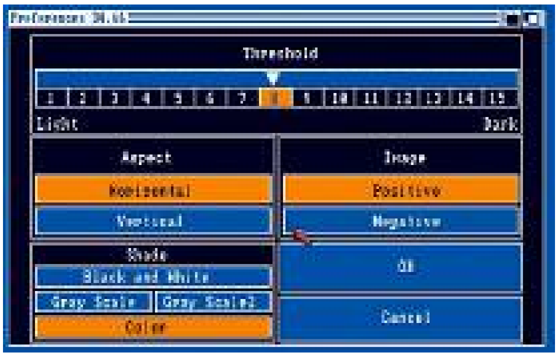
3.1.3.4 Printer



Displays the printer preferences screen.

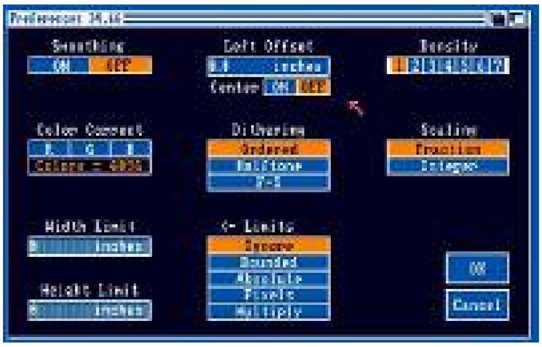
Parallel or Serial - Sets whether printer is connected via parallel or serial port.
Printer Name - Selects a printer driver (see InstallPrinter to view new drivers)
Paper Size - Sets paper format for printer
Length - Sets number of lines down the page
Paper Type - Sets type of paper, fanfold for dot matrix, Single for lasers or inkjets etc
Quality - Sets quality of output, either Draft or Letter
Left/Right Margin - Sets number of characters from left or right margins
Pitch - Sets size of fonts used for text
Spacing - Sets number of lines per inch for text
Graphic 1 - Graphic settings page 1
Graphic 2 - Graphic settings page 2
OK - Accept new printer settings
Cancel - Cancel and forget printer settings

<u>Graphic1</u>



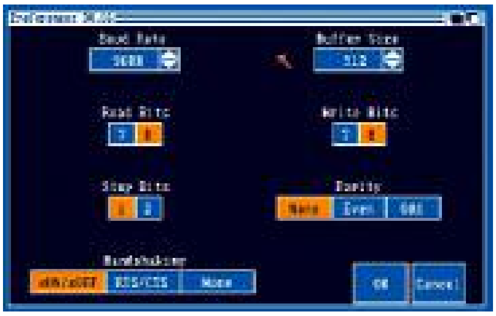
Threshold - Determines how dark or light the picture is
Aspect - Determines whether graphics is printed Horizontally (Landscape) or Vertical (Portrait)
Image - Determines if graphics are to be shown as is (Positive) or reverse graphics (Negative)
Share - Determines colour scheme: B&W, Gray or Colour

Graphic2



Smoothing - Determines if graphic edges are to be smoothed out or not
Left Offset - Sets number of inches from left side of page
Center - Sets whether image is to be centered on the page
Intensity - Determines resolution of the image (1 is low, 7 is high)
Color Correct - Adjusts colour levels in RGB terms for printing
Dithering - Sets dithering mode of pixels to get best image quality
Scaling - Sets how the image is to be scaled to fit page
Width Limit - Sets maximum width of the image in inches
Height Limit - Sets maximum height of the image in inches
Limits - Sets type of limits to apply to the image

3.1.3.5 Serial



Displays the serial preferences screen. These preferences can apply to serial printers and modems or even peer to peer networks.

Baud Rate - Specifies the speed of the connection from 110 to 31325
Buffer Size - Size of serial buffer in bytes
Read Bits - Number of bits for data (7 for ASCII, 8 for Binary)
Write Bits - Number of bits for writing data
Stop Bits - Number of bits to stop stream of data
Parity - Number of parity bits to check for error checking
Handshaking - Determine method of communication between systems

3.1.4 Hidden Drawers

C	- Stores AmigaDOS commands
Devs	- Stores keymap files, printer drivers and other device drivers and the preferences file.
Fonts	- Stores bitmap font files. A .font file for each typeface and a file for each font size.
L	- Stores device handlers
Libs	- Stores Amiga operating system libraries
S	- Stores batch or script files
T	- Temporary files drawer

3.1.5 Other Drawers

Expansion - Stores special device drivers for expansion peripherals
Empty - Duplicate this drawer to create new drawers.
Trashcan - Stores files ready to be deleted.

3.2 Extras Disk

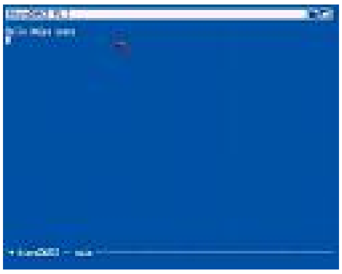
AmigaBASIC

BASIC Programming Language interpreter

Here you have a simple BASIC (Beginners All Purpose Symbolic Interpreted Code) programming language. You type commands such as CHDIR, LOAD, SAVE, RUN, LIST, SYSTEM in the main ‘Basic’ window and view or edit any programs in the ‘List’ window. You can also load, save etc from the menus.
Example programs can be found in the Basic Demos folder.

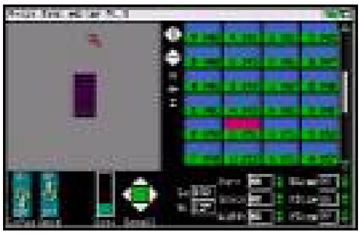
3.2.1 Tools Drawer

3.2.1.1 MEmacs



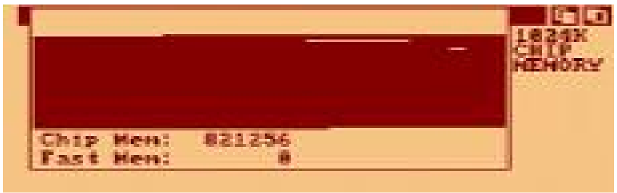
MicroEmacs is a powerful text editor for the Amiga which type can type documents and so on. Includes load, save, find and replace, multiple windows, macros and so on. See menus for commands.

3.2.1.2 Fed



Fed is a bitmap font editor which you can create bitmap fonts for use on the Amiga. For example, in the screen above there is the editor part on the top left, the character set shown on the top right and various options on the bottom to edit fonts.

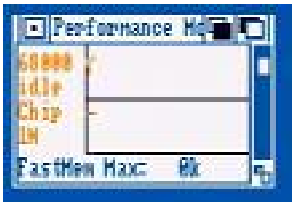
3.2.1.3 Freemap



Chip Mem: 821256
Fast Mem: 0

Freemap displays the amount of free memory available as a map on a separate screen at the bottom of the display. Useful for occasions when experiencing low memory problems. To quit select 'Please' from the 'Quit' menu.

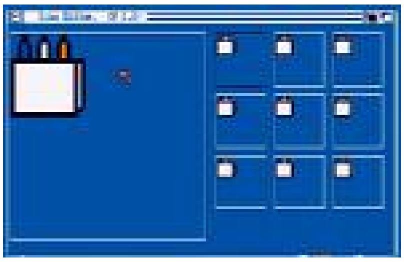
3.2.1.4 PrefMon



FastMem Max: 0k

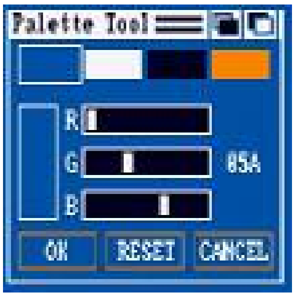
This is a real time prformance monitor of CPU and memory usage over time displayed at a graph in separate window sections. Useful to see what happens when running given programs or processes. You can select how often samples are taken from the PM menu.

3.2.1.5 IconEd



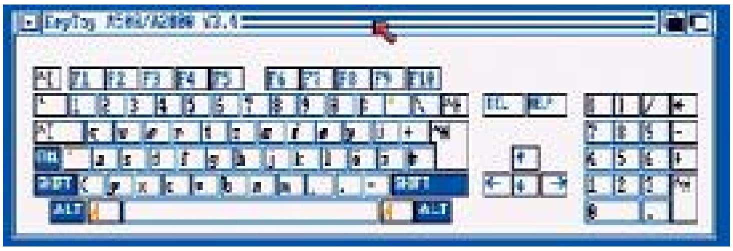
The Icon editor can be used to edit existing or create new type of icons for disks, drawers, programs (tools), data files (projects) and the Trashcan. For workbench to display files, each file must have an icon file with it. You can copy icon files, either via the Shell or via Icon edit and save it as a new info file. See menus for various options to make your icons.

3.2.1.6 Palette



This program can change Workbench colour scheme. The colour scheme only lasts until the next reboot so you can save the changes only temporarily apply them to the workbench screen.

3.2.1.7 KeyToy2000



Displays the current keyboard characters using the current keymap. Pressing the CTRL, SHIFT or ALT keys will display the alternative keys when those keys are used.To change keymaps you must use the Setmap command and select a different mapping.

3.2.1.8 IconMerge

This program can create dual-image icons from two icon files or split an existing icon file into two files. This is useful if you wish to create icons with different images when an icon is selected or not.

3.2.1.9 Terminal

This program can be used to connect to other computers or use a modem to connect to Bulletin Boards over the telephone. Its appears in WB 1.3D and is in German.

3.2.2 PCUtil

PCCopy	- Allows Amiga to copy files from PC disks
PCFormat	- Allows Amiga to format PC diskettes
ToPCCopy	- Allows Amiga to copy files to PC disks
Read Me	- Text file on using PCUtils

3.2.3 Other Drawers

BasicDemos	- Basic program listings
FD1.3	- Contains OS Function Descriptions for AmigaBASIC
Fonts	- More fonts
Devs	- Contains all keymap files and printer drivers

4 AmigaDOS Commands

Amiga DOS commands can only be used on a CLI or Shell window where commands can be typed in to perform various functions. All these commands have to be run from disk.

4.1 Command format

To display the information required by a command type a '?' after it. Note, after each argument there is a slash and a character. This specifies the kind of argument required such as an option or a name or a value.

/A - Argument must be supplied
/F - Must be final or last argument
/K - Keyword must be entered with a parameter
/M - Multiple arguments may be supplied
/N - A Number is required
/S - Switch or option. Supply this function to activate it.

4.2 Error messages

If you enter a command and comes back with an error message, typing WHY will display more information about it. Use the FAULT command to display the error message that applies to a error code.

Typical error messages are:

Code	Description	Solution
103	Insufficient free store .	Free up some memory by quiting other programs
105	Task table full	Shutdown some programs
120	Argument line invalid or too long	Check command arguments using `?`
121	File is not an object module	Try setting 'e' or 's' protection flag.
122	Invalid resident library during load	
202	Object in use	Exit program, unassign assign or close directory windows.
203	Object already exists.	Cannot move the program as it exists elsewhere. Delete the original first
204	Directory not found	Retype directory name
205	Object not found	Retype file name
206	Invalid window description	Re-enter correct window description for NEWCLI or NEWSHELL
209	Packet request type unknown	
210	Stream name component invalid	
211	Invalid object lock	Recheck filename
212	Object not of required type	Recheck file
213	Disk not validated	Wait until disk is validated first.
214	Disk write-protected	Ensure write tab is closed.
215	Rename across devices attempted	Use copy instead

216	Directory not empty	Must delete its contents first. Use DELETE ALL.
218	Device (or volume) not mounted	Check device name and reinsert disk
219	Seek failure	Check position in file
220	Comment too big	Use shorter description in FILENOTE
221	Disk full	Delete some files or use a new disk.
222	File is protected from deletion	Enable Delete flag with Protect
223	File is write protected	Enable Write flag with Protect
224	File is read protected	Use PROTECT to set 'r' flag.
225	Not a valid DOS disk .	Disk is unformatted or a protected game disk
226	No disk in drive	Use correct device name or insert disk
232	No more entries in directory	Directory is full, delete files or move files into new sub-dirs.

There are another set of error messages called `Guru Meditation' which is displayed when the computer crashes. It consists of two numbers. the first is the error id and second the address of the task:

ssggeeee aaaaaaaa

ss = Subsystem Id, gg = general error, eeeee = specific error

The most common ones are CPU traps:

00000002	Bus Error (memory doesn't exist)
00000003	Address error (usually odd address access)
00000004	Illegal instruction
00000005	Divide by zero
00000006	CHK instruction (Check register against boundaries)
00000007	TRAPV instruction (Trap on overflow)
00000008	Privilege violation
00000009	Trace (debugging)
0000000A	Op Code 1010 (unimplemented instruction)
0000000B	Op Code 1111 (unimplemented instruction)

Exec:

81000005	Corrupted memory list
81000009	Memory freed twice

4.3 Volume/Drawer/File names

Names can be up to 31 characters long and may contain the following characters:

Letters A-Z (upper or lower)
Digits 0-9
Other characters EXCEPT the colon ':' or slash '/'.
Spaces can be used but can be confusing use a dash '-' or
underline '_' instead.

4.4 Volumes, Drawers and Assigns

A volume is the name given to a disk and may be referred to
by its name followed by a colon ':' e.g. DIR DataDisk:

A drawer or directory is used to organise files together and they are usually used in conjunction with a forward slash to separate the names
e.g. DIR DF0:Devs/Printers

An assign is a special shortcut name to a volume or a drawer and is created using the ASSIGN command.
e.g.

```
    TYPE S:Startup-sequence     is the same as
    TYPE Workbench1.3:S/Startup-Sequence
```

4.5 Devices

A device is a special name given to a particular peripheral or an Amiga DOS feature. Typical devices are:

DF0:	- Internal floppy drive
DF1:	- External floppy drive
DH0:	- Hard disk (A590)
RAM:	- RAM disk
RAD:	- Recoverable RAM disk (from a reboot)
CON:	- Console device (CLI)
NEWCON:	- New Console device (Shell)
SER:	- Serial port
PAR:	- Parallel port
PRT:	- Printer port
AUX:	- Auxillary device
PIPE:	- Pipe device
CD0:	- CD-ROM (A570)

4.6 The Commands

Addbuffers
Format: Addbuffers Drive/A Buffers/N
Purpose: Add memory to floppy disk buffers to improve speed
Example: Addbuffers df0: 10

Ask
Format: Ask Prompt/A
Purpose: Asks a question in a script (reply y/n). Use IF WARN to test for a yes.
Example: Ask "Are you sure? Y/n"

Assign
Format: Assign Name Target/M LIST/S
Purpose: Assign names to directory paths.
Example: Assign PPaint: DH0:PPaint

Avail
Format: Avail CHIP/S FAST/S TOTAL/S FLUSH/S
Purpose: Memory availability (WB1.3)
Example: Avail TOTAL

Binddrivers
Format: BindDrivers
Purpose: Load device drivers for old hardware.
Example: BindDrivers

Break
Format: Break Task/A/N ALL/S C/S D/S E/S F/S
Purpose: Stop a program (see Status)
Example: Break 2 C

CD
Format: CD Dir/A
Purpose: Change directory
Example: CD Text_Files

ChangeTaskPri
Format: ChangeTaskPri Pri=Priority/A/N Process/K/N (WB1.3)
ChangeTaskPri Pri/A (WB1.2)
Purpose: Changes program's execution priority (see Status)
Example: ChangeTaskPri 2 1

Copy
Format: Copy From/M To/A ALL/S QUIET/S BUF=BUFFER/K/N CLONE/S DATES/S NOPRO/S COM/S (WB1.3)
Copy From/M To/A ALL/S QUIET/S (WB1.2)
Purpose: Copy files (with/without protection bits)
Example: Copy mydoc.txt docs: Dates

Date
Format: Date Time Date To=Ver/K
Purpose: Display/change date or time
Example: Date To=date.txt

Delete
Format: Delete File/M/A ALL/S QUIET/S
Purpose: Delete files and/or directories
Example: Delete OldFilesDir All

Dir
Format: Dir Dir Opt/K ALL/S DIRS/S INTER/S (WB1.3)
Dir Dir Opt/K (WB1.2)
Purpose: List contents of a directory
Example: Dir S: Files

Diskchange
Format: Diskchange Device/A
Purpose: Tell Amiga DOS that a disk has been swapped in a non-auto-sensing disk drive
Example: Diskchange DF2:

Diskcopy
Format: Diskcopy From/A To/A Name/K NOVERIFY/S MULTI/S
Purpose: Duplicates disks
Example: Diskcopy df0: to df1:

DiskDoctor
Format: DiskDoctor Device/A
Purpose: Repair damaged files on a disk
Example: DiskDoctor DF0:

Echo
Format: Echo String/M NOLINE/S (WB1.3)
Echo String (WB1.2)
Purpose: Display text in a script or part of a string of text
Example: Echo "Enter your name: " NoLine

Ed
Format: ED From/A Size/N
Purpose: Edit text files (screen editor)
Example: ED S:User-Startup

Edit
Format: Edit From/A To With/K Ver/K Opt/K
Purpose: Edit text files (line editor)
Example: Edit s:User-Startup With Ed-Script

Else
Format: Else
Purpose: Used with IF, if not true do these commands
Example: If Exists s:User-Startup
```
        Say "User Startup already exists"
        Else
        Ed s:User-Startup
        Endif
```

Endcli
Format: Endcli
Purpose: Close CLI console window
Example: Endcli

Endif
Format: Endif
Purpose: End a IF..ELSE..ENDIF block
Example: If $Age GT 65
```
        Say "You are a pensioner!"
        Endif
```

Endskip
Format: Endskip
Purpose: Stops a SKIP branch (for debugging) (WB1.3)
Example: Endskip

Eval
Format: Eval Value1/A Op Value2/M To/K LFormat/K
Purpose: Evaluate a simple arithmetic expression (WB1.3)
Example: Eval $count + 1 to env:count

Execute
Format: Execute File/A
Purpose: Execute a script or batch file
Example: Execute PCD

Failat
Format: Failat RCLIM/A
Purpose: Change failure limit for scripts (0,5,10,20)
  (0 - Ok, 5 - Warn, 10 - Error, 20 - Fail)
Example: Failat 20

Fault
Format: Fault Error/N/M
Purpose: Display error message from a code
Example: Fault 221
```
        "Fault 221: Disk is full"
```

FF
Format: FF -o -n
Purpose: Activate Fast Font program (WB1.3)
Example: FF

FileNote
Format: Filenote File/A Comment
Purpose: Attach a comment to a file (see LIST)
Example: FileNote List.txt "A list of people's names"

Format
Format: Format Device=Drive/K/A Name/K/A NOICONS/S FFS/S NOFFS/S (WB1.3)
        Format Drive/K/A Name/K/A NOICONS/S (WB1.2)
Purpose: Initialises Amiga disks (in System drawer)
Example: Format Drive DF0: Name "Docs" NOICONS

Getenv
Format: Getenv Name/A
Purpose: Return contents of a global environment variable (WB1.3)
Example: Getenv Workbench

Iconx
Format: Iconx
Purpose: Enables a script to be run from Workbench. The script
        must have a Project .info file and the Default Tool
        set to c:IconX to run it. (WB1.3)
Example: Iconx

If
Format: If NOT/S WARN/S ERROR/S FAIL/S EQ/K EXISTS/K
Purpose: If condition is true execute following commands until
        an ELSE or ENDIF occurs.
Example: IF WARN, IF $count EQ 10, IF EXISTS C:List

Info
Format: Info Device (WB1.3)
        Info (WB1.2)
Purpose: Displays information about disks
Example: Info DH0:

Install
Format: Install Device/A NOBOOT/S CHECK/S (WB1.3)
        Install Device/A (WB1.2)
Purpose: Saves a bootblock to a disk. If you have only one drive type INSTALL ? and swap disks and then type DF0: and press RETURN to save bootblock.
Example: Install DF0:

Join
Format: Join File/M/A AS=TO/K/A
Purpose: Join two or files together.
Example: Join File1 File2 File3 as FileFull

Lab
Format: Lab Label
Purpose: State a Label to Skip to in a script
Example: Lab PROC2

List
Format: List Dir/M P=PAT/K KEYS/S DATES/S NODATES/S To/K Sub/K
        Since/K Upto/K QUICK/S BLOCK/S NOHEAD/S FILES/S DIRS/S
        LFORMAT/K (WB1.3)
        List Dir/M P=PAT/K KEYS/S DATES/S NODATES/S To/K Sub/K
        Since/K Upto/K QUICK/S (WB1.2)
Purpose: List file details in a directory
Example: List Workbench: Block Dates Since=01-Jul-00

LoadWB
Format: Loadwb
Purpose: Display workbench screen at startup
Example: LoadWB

Lock
Format: Lock Drive/A ON/S OFF/S Passkey
Purpose: Write protects any HD partition (WB1.3)
Example: Lock DH1: ON MyPass

MakeDir
Format: Makedir Name/A
Purpose: Create a directory.
Example: MakeDir DH1:Docs

Mount
Format: Mount Device/M From/K (WB1.3)
        Mount Device/A (WB1.2)
Purpose: Loads and mount a device
Example: Mount DEVS:CD0

Newcli
Format: Newcli Window From
Purpose: Open a new CLI console (CON) window
        Window format: CON:X/Y/Width/Height/Name
Example: Newcli CON:0/10/400/200/Shell

Newshell
Format: Newshell Window From
Purpose: Open a new CLI console (CON) window (WB1.3)
        Window format: CON:X/Y/Width/Height/Name
Example: Newshell CON:0/10/400/200/Shell

Path
Format: Path Path/M ADD/S SHOW/S RESET/S
Purpose: Set or change the program search path
Example: Path SC:C Add

Prompt
Format: Prompt Prompt
Purpose: Change the prompt text (Codes: %R = last return code,
        %S = Current path, %N = CLI Number)
Example: Prompt "%N.%S> "

Protect
Format: Protect File/A Flags ADD/S SUB/S (WB1.3)
        Protect File/A Flags (WB1.2)
Purpose: Change the protection flags of a file where r=read, w=write, e=execute, d=delete, s=script, p=pure)
Example: Protect Editor rwed

Quit
Format: Quit RC/N
Purpose: Quit a script with optional return code
Example: Quit 5

Relabel
Format: Relabel Drive/A Name/A
Purpose: Change name of a disk
Example: Relabel DF0: "Documents"

RemRAD
Format: RemRAD
Purpose: Removes contents of recoverable ram disk
Example: RemRAD

Rename
Format: Rename From/A/M To=As/A QUIET/S
Purpose: Rename or move one or more files
Example: Rename MyDoc as MyDoc.doc

Resident
Format: Resident Name File ADD/S REPLACE/S REMOVE/S
Purpose: Makes a CLI command memory resident (WB1.3)
Example: Resident C:Dir

Run
Format: Run Command/F
Purpose: Execute a program in the background (multi-task)
Example: Run Printfiles Doc1

Search
Format: Search From/M Search/A ALL/S NONUM/S QUIET/S QUICK/S
    FILE/S (WB1.3)
    Search From Search/A ALL/S (WB1.2)
Purpose: Search for a string in a file or a file on a disk
Example: Search SYS: "List" File All

SetClock
Format: Setclock OPT LOAD/S SAVE/S
Purpose: Load or set the date and time from battery-backed clock
Example: Date 27-Nov-00 10:00, SetClock Opt Save

SetDate
Format: Setdate File/A Date Time
Purpose: Set date of a file or directory (see List)
Example: Setdate DEVS: Monday 10:30

Setenv
Format: Setenv Name String/F
Purpose: Change the contents of a environment variable (WB1.3)
Example: Setenv Workbench "1.3"

SetMap
Format: Setmap Keymap/A
Purpose: Sets keyboard layout
Example: Setmap UK

SetPatch
Format: Setpatch
Purpose: Prevents Guru Meditation after a Recoverable alert (WB1.3)
Example: Setpatch

Skip
Format: Skip Label
Purpose: Jump to a label in a script (see Lab).
Example: Skip Lab2

Sort
Format: Sort From/A To/A Colstart/K
Purpose: Sort file into another file
Example: Sort Names To SortedName Colstart=10

Stack
Format: Stack Size/N
Purpose: Change/display amount of stack space in bytes
Example: Stack 10000

Status
Format: Status Process/N FULL/S TCB/S CLI=ALL/S COM=COMMAND/K (WB1.3)
        Status Process/N FULL/S TCB/S CLI=ALL/S (WB1.2)
Purpose: Display status of running programs. TCB is Task Control Block
Example: Status 2 Full

Type
Format: Type From/A/M TO/K OPT/K HEX/S NUMBER/S (WB1.3)
        Type From/A/M TO/K OPT/K (WB1.2)
Purpose: Display contents of text file on screen and binary files using
        HEX with optional line numbers.
Example: Type C:DIR HEX

Version
Format: Version
Purpose: Display workbench and kickstart version information
Example: Version

Wait
Format: Wait Time/N SEC=SECS/S MIN=MINS/S UNTIL/K
Purpose: Wait for a specified period of time
Example: Wait Until 12:00

Why
Format: Why
Purpose: Display reason why a previous command failed
Example: Why

4.7 AmigaBASIC Command List

ABS(x) - Absolute value
AREA [STEP](x,y) - Define an area
AREAFILL [mode] - Fill an area
ATN(x) - ArcTangent
BEEP - Make a sound
BREAK ON|OFF|STOP - Allow Break in code
CALL name[(args)] - Call subprogram
CDBL(x)
CHAIN [MERGE] file [,[expr][,|ALL][,DELETE range]] - Load and run a program
CHDIR string - Change directory
CHR$(i) - ASCII code to character
CINT(x) - Convert value to integer with rounding
CIRCLE [STEP](x,y),radius [,color [,start,end [,aspect]]] - Create circle or arc
CLEAR [,BasicData][,stack] - Clear memory
CLNG(n) - Convert to long integer
CLOSE [[#]filenum[,[#]filenum...]] - Close files
CLS - Clear screen
COLLISION ON|OFF|STOP - Enable/disable sprite collisions
COLOR [fore][,back] - Set foreground/background colour
COMMON var-list - Define common variables
CONT - Continue
COS(x) - Cosine
CSNG(x) - Convert to single precision value
CSRLIN - Return print line in current window or screen
CVI(2-byte string) - Convert string to integer
CVL(4-byte string) - Convert string to long integer
CVS(4-byte string) - Convert string to single precision value
CVD(9-byte string) - Convert string to double precision value
DATA constant-list - List data items
DATE$ - Current date
DECLARE FUNCTION id[(param list)] LIBRARY - Start a multi-line function
DEF FNname[(param list)]=func-def - Define a single-line function
DEFDBL letter-range - Define Double-precision variables
DEFINT letter-range - Define Integer variables
DEFLNG letter-range - Define Long Integer variables
DEFSNG letter-range - Define Single-precision variables
DEFSTR letter-range - Define String variables
DELETE [line][-line] - Delete range of Basic statements
DIM [SHARED] var-list - Define arrays
END - End of program
EOF(fileno) - End of file
ERASE array-var-list - Delete arrays
ERL - Line no. of error
ERR - Error number
ERROR int-expr - Cause an error
EXP(x) - Exponent
FIELD [#]fileno, fieldwidth AS string-var.. - Define a Field in a file
FILES [string] - List files in current dir (to optional file)
FIX(x) - Truncated integer component of value
FOR var=x TO y [STEP z] - Counted loop
  NEXT [var][,var...]
FRE ({-1|-2|x}) - Free memory
GET [#]|filenum|[,rec-num] - Get record from file
GET (x1,y1)-(x2,y2),array-name - Get area into array
GOSUB lineno - Call subroutine
  RETURN [line]

GOTO lineno - Jump to another part of program
HEX$(x) - Hexadecimal value of number
IF expr GOTO line [ELSE stat] - If expression true goto line else run other statement
IF expr THEN stat [ELSE stat] - If expression true run one statement else run another
INKEY$ - Input a character from keyboard
INPUT [;][prompt;] var-list - Input values with optional prompt string
INPUT$(x[,[#]fileno]) - Input x characters from keyboard or file
INPUT #fileno,var-list - Input values from a file
INSTR([i],x$,y$) - Find position of y$ in x$ from postion i
INT(x) - Integer value
KILL filespec - Delete a file
LBOUND (array-name[,dim]) - Define lower boundary of an array
UBOUND (array-name[,dim]) - Define upper boundary of an array
LEFT$(x$,i) - Get first i characters of x$
LEN(x$) - Length of string
[LET] variable = expr - Assign value of expression to variable
LIBRARY "filename" - Open an AmigaOS library
LIBRARY CLOSE - Close an AmigaOS library
LINE [[STEP](x1,y1)]-[STEP](x2,y2),[color][,b[f]] - Draw a line
LINE INPUT [;]["prompt";]string-var - Input whole line into string
LINE INPUT #fileno, string-var - Input whole line from file
LIST [line] - List statements in program
LIST [line][-[line]],"filename" - List statements in file
LLIST [line][-[line]] - List statements to printer
LOAD [filespec[,R]] - Load file
LOC(fileno) - Location in file
LOCATE [line][,column] - Locate printing position on window or screen
LOF(fileno) - Length of file in bytes
LOG(x) - Logarithm
LPOS(x) - Line position
LPRINT [expr-list] - Print values to printer
LPRINT USING string-expr;expr-list - Print values according to format instructions
LSET string-var=string-expr - Set field with left justification
MENU menu-id,item-id,state[,title-string] - Define menu item
MENU RESET - Clear menu items
MENU ON|OFF|STOP - Enable/disable menus
MERGE filespec - Load and merge basic program
MID$(x$,n[,m])=y$ - Replace characters in middle of string
MKI$(short-int-expr) - Create 2-byte string of integer
MKL$(long-int-expr) - Create 4-byte string of long integer
MKS$(single-expr) - Create 4-byte string of single-precision value
MKD$(double-expr) - Create 8-byte string of double-precision value
MOUSE(n) - Status of mouse buttons, position
MOUSE ON|OFF|STOP - Disable/enable mouse events
NAME "old-file" AS "new-file" - Rename a file
NEW - Clear memory including program
NEXT [var[,var...]] - End of FOR loop
OBJECT.AX object-id,value - Position sprite in x
OBJECT.AY object-id,value - Position sprite in y
OBJECT.CLIP (x1,y1)-(x2,y2) - Clip object`s size
OBJECT.CLOSE [obj-id[,obj-id...]] - Remove sprite
OBJECT.HIT obj-id, [MeMask][,HitMask] - Set sprite`s collision zone
OBJECT.PLANES obj-id[,plane-pick][,plane-on-off] - Set sprites colour planes
OBJECT.PRIORITY obj-id, value - Set sprite`s priority
OBJECT.SHAPE obj-id, definition - Set sprite`s shape
OBJECT.SHAPE obj-id, obj-id2 - Copy sprite`s shape
OBJECT.START [obj-id[,obj-id...]] - Start sprite motion
OBJECT.STOP [obj-id[,obj-id...]] - Stop sprite motion
OBJECT.VX obj-id,value - Set sprite x velocity

OBJECT.VY obj-id,value - Set sprite y velocity
OBJECT.X obj-id, value - Set sprite x position
OBJECT.Y obj-id,value - Set sprite y position
OCT$(x) - Octal value
ON BREAK GOSUB label|0 - If CTRL+C detected goto line or stop
ON COLLISION GOSUB label|0 - If collision detected goto line or stop
ON ERROR GOTO line - If error occurs goto line
ON expr GOSUB line-list - On value of expression goto one of lines from list
ON expr GOTO line-list - On value of expression goto one of lines from list
ON MENU GOSUB label|0 - If menu item selection goto line
ON MOUSE GOSUB label|0 - If mouse event occurs goto line
ON TIMER(n) GOSUB label|0 - On timer event goto line
OPEN mode,[#]fileno,filespec,[,buffer] - Open a file
OPEN filespec [FOR mode] AS [#]fileno [LEN=buffer] - Open a file
OPTION BASE n - Specify 0|1 for first item in array
PAINT [STEP](x,y)[,paint-color][,border-color]] - Paint an area
PALETTE color-id,r,g,b - Create colour
PATTERN [line-pat][,area-pat] - Define line/area patterns
PEEK(address) - Get contents of memory address
PEEKL(address) - Get long value from memory
PEEKW(address) - Get word value from memory
POINT(x,y) - Get colour id of pixel point
POKE addr,value - Set a memory address
POKEL addr,value - Set long value in memory
POKEW addr,value - Set word value in memory
POS(x) - Returns print column
PRESET [STEP](x,y)[,color] - Set pixel colour
PRINT [expr-list] - Print values to screen
PRINT USING string-expr;expr-list - Print values to screen using format
PRINT #fileno,[USING string-expr;]expr-list - Print values to file
PSET [STEP](x,y)[,color] - Set pixel colour
PTAB(x) - Set print tab position
PUT [#]fileno[,record-no] - Store record in file
PUT [STEP](x,y),array[(index[,index...])][,action-verb] - Place object on screen
RANDOMIZE [expr]|TIMER - Set initial random seed
READ var-list - Read DATA values
REM remark - Place program documentation
RESTORE [line] - Restart from DATA line
RESUME [{0|NEXT|line}] - Resume from error
RETURN [line] - Return from subroutine
RIGHT$(x$,i) - Get right most characters from string
RND[(x)] - Random number
RSET string-var=string-expr - Set string, right justified
RUN [line]|filename[,R] - Run from line or program
SADD(string expr) - String address
SAVE [filename [{,A|,P|,B}]] - Save basic program
SAY "string" [,mode-array] - Say something through speech synthesis
SCREEN screen-id,width,height,depth,mode - Set up a screen of given size and depth
SCREEN CLOSE screen-id - Close a screen
SCROLL (x1,y1)-(x2,y2),delta-x,delta-y - Scroll a portion of the screen
SGN(x) - Sign of value
SHARED var-list - Define list of shared variables
SIN(x) - Sine of value
SLEEP - Goto sleep, allow for multi-tasking
SOUND freq,duration[,[vol][,voice]] - Create a sound note
SOUND WAIT|RESUME
SPACE$(x) - Return x space characters
SPC(x) - Use with Print for x Space characters
SQR(x) - Square root of x

STICK(n)	- Status of joystick
STOP	- Stop program
STRIG(n)	- Status of joystick button
STR$(x)	- String value of x
STRING$(i,j)	- Return i number of j characters
SUB subprog-namep[(param-list}][STATIC]	- Start of subprogram
END SUB	
EXIT SUB	
SWAP var1,var2	- Swap contents of variables
SYSTEM	- Exit to Workbench
TAB(x)	- Print x tab characters
TAN(x)	- Tangent
TIME$	- Current time
TIMER ON|OFF|STOP	- Enable/disable timer events
TRANSLATE$("string")	- Translate string into phonome text
TRON	- Trace on
TROFF	- Trace off
UBOUND(array [,dim])	- Get upper boundary of array
UCASE$(string)	- Convert string to upper case
VAL(x$)	- Numeric value of string
VARPTR(var)	- Address of variable
WAVE voice,wave-definition	- Generate waveform for sound
WIDTH device,[size][,print-zone]	- Define width of print zone for device
WIDTH #fileno,[size][,print-zone]	- Define width of file
WIDTH [size][,print-zone]	- Define width
WIDTH LPRINT [size][,print-zone]	- Define width of printer
WINDOW window-id [,[title][,[rectangle][type[,][,screen-id]]]]	- Create a window
WINDOW CLOSE window-id	- Close a window
WINDOW OUTPUT window-id	- Set default output window
WINDOW(n)	
WRITE [expr-list]	- Write values to screen
WRITE #fileno,expr-list	- Write values to file

More information is in the AmigaBasic manual. Equivalent information is available with HiSoft Basic and ACE Basic. AmigaBasic does not run well on WB2 or above so replace it with more advanced Basic such as Hisoft Basic, ACE, Amos or Blitz Basic.

5. Common Problems

5.1 How do I retreive a file I accidently deleted?

You need to use a Disk Recovery program such as DiskDoctor, FixDisk or Disksalv to scan and recover the file. You cannot retrieve a file that has been deleted and the disk written to later on as it will most likely been overwritten!

5.2 How can I view hidden file from Workbench?

There are a number of methods although the best is to use a file manager program which can see all files and you can copy, move, delete and view files very easily. Other than that you need to resort to using the Shell.

To make a file viewable, you need to create a .info file for it, you can either copy a .info file to the new file e.g. copy oldfile.info newfile.info (of the same type) or use IconEdit to draw and save the file icon.

5.3 How can I create a bootable disk?

The simplest method is to Duplicate your Workbench disk, delete the Utilities and the System files (if not required), copy the program to the disk and then edit the s:startup-sequence to run your program. This method also guarentees that all the necessary files are also on the disk.

The minimum required to make a disk bootable is to use the Install command to save a bootblock to the disk. Create a directory called S and create a startup-sequence file with the name of the program and save that in the S directory and copy your program to the disk. To use install with one disk drive, type INSTALL ? then swap disks and type DF0: and press RETURN.

5.4 How can I speed up disk access when using the shell?

The shell is a very powerful feature but can be limiting and annoying when working from floppy. You can make common commands run from memory by using the Resident command which allows the Amiga to use commands from memory instead of loading them from disk. For example, Resident C:Dir pure will make the DIR command resident. You can make this a permanent feature by modying the S:Shell-Startup file and insert these resident commands.

Also, it is a good idea to do all work on the RAM: disk if you have plenty of memory instead of saving to/from disk.

5.5 Why do some icons, when double-clicked, don't load up?

There are two reasons for this. First, if a program is deleted but its .info file is left behind then the program or file may seen to still exist. To rectify this, the program should be re-copied back.

Second, a lot of Project icons have A Default Tool set. This default tool loads up the program and the data file. There are many tools, particularly text and graphics viewers. If you try to run a file with a Default Tool that you don't have then the following message will appear: 'Unable to open the tool <file>' If this happens, select the icon and choose Info from the Workbench menu and change the Default tool to the one you do have. For example, if it’s a text file, change it to 'More' (Workbench's Text Viewer).

Beyond Workbench 1.3

Workbench 1.3 is one of the oldest and least usable of the Amiga's OS. Over the years, plenty of utilities have been released to improve some of the oversights of WB 1.3.

The following programs and packages could be installed to improve your workbench. Some programs have been upgraded to WB2 or WB3 so check requirements first.

a) SID (and other directory utilities)
SID was a very popular directory utility to organise files on your floppy disks or hard disk. A variant called MessySID allowed access to PC disks too. (Aminet/util/dir)

b) DMouse.
A Mouse accelerator which included other options including screen saver, pointer blanker, auto window activation, programmable keys, window control etc. Essential! (Aminet/util/cli)

c) FixDisk or DiskSalv v1.x
Essential disk repair programs for fixing floppy or hard disk problems. Later versions only work on WB2 or better. (Aminet/disk/salv/)

d) PowerPacker
Useful packing program with good viewers such as PPShow, PPMore to read/view normal or packed files.

e) ARP or Amiga Replacement Project
A replacement for WB 1.3s older C commands and comes with the arp.library for standard file requesters before ASL and REQTOOLS appeared. Difficult to find these days.

f) Magic Workbench.
Yes, the ultimate icon package IS available for WB1.3 users. Look out for MagicWB1.3.lha in Aminet for useful WB 1.3 tools. (Aminet/pix/mwb)

g) MSH (or MessyDOS) and MultiDOS.
This was available before CrossDOS appeared as standard. You will be able to read/write PC disks. (Aminet/misc/emu)

h) Virus Checkers
WB 1.3 users have a big problem in this area as most new virus checkers are for WB2 or better unless you know different? VirusX and Virus Checker v6.xx will remove older viruses though. (Aminet/util/virus)

i) HDClick - Program Launcher.
WB1.3 does not have a Tools menu like WB2 or better, so other types of program launchers are available to quickly start your programs such as HDClick 2.01 (the last one before becoming WB2 only).

Adding a Hard Disk to A500

Unfortunately, the A500 does NOT come with a Hard Disk controller, so you cannot just plug in any IDE or SCSI disk unless you have one.

The original hard disk was the A590 by Commodore which included a 20Mb XT Hard disk, this can be upgraded to a large SCSI hard disk later. This device plugs into the Expansion port on the left side of the A500. Other manufacturers including GVP, Dataflyer, have developed a hard disk controller and disk combination for both SCSI and IDE types of hard disk.

As for hard disk sizes, you can have up to 4Gb Drives but beyond that, the file system needs to be upgraded to utilise beyond the 32 bit limit of Workbench otherwise the data on the hard drive can become corrupted. Installing Kickstart 3.1 and Workbench 3.1 plus a file system such as SFS or PFS can overcome this limitation.

Installing Workbench onto a Hard Disk

To install Workbench 1.3 onto a Hard disk, you need to partition and format the hard disk. The standard tool is HDToolbox, but it is not always supplied, see the equivalent tools and documentation for your Hard disk.

Once the Hard disk has been set up and partitioned and marked as bootable you can install Workbench. Workbench 2.0 and later has an Install disk with scripts to do this for you. For Workbench 1.3 you have to do this manually by copying the entire contents of the Workbench and Extras disk to your hard disk.

For example,

1. Open a Shell to get a command prompt
2. Insert the Workbench disk in DF0:
3. Type COPY DF0:#? To DH0: ALL CLONE which will copy all files from the floppy in DF0: to the harddisk partition DH0:.
4. Remove the Workbench disk and insert the Extras disk
5. Type COPY DF0:#? To DH0: ALL CLONE to copy all the files to your hard disk.

For most application programs, except games, you can install them to your hard disk as well using a supplied install program or just by drag and dropping the files from the disk into a new drawer (also known as a directory or folder) on your hard disk.

Adding more Memory to the A500

The Amiga memory is split into Chip RAM (memory used by the custom chipset to store graphics and sound) and Fast RAM (memory which is used only by the Processor).

The Amiga A500 can come with 512K or 1M of memory. You can increase Chip RAM up to the maximum of 2Mb and Fast RAM up to 7Mb or 8Mb (depending if you have 1Mb or 2Mb of Chip RAM) to a total of 9Mb.

You can increase RAM by adding memory via a new memory board into the expansion slot underneath the Amiga behind a trapdoor or via the Expansion Port on the left hand side, sometimes as part of another device such as the A590.

When increasing Chip RAM you must also upgrade the Agnus chip which is used to address the Chip RAM area to a new version that can handle 1M or 2Mb or Chip RAM.

See documentation with a board or device on how to install extra memory.

Upgrading the Processor

The standard 68000 7MHz processor in the Amiga A500 is quite slow and cannot cope with later applications and games. The processor is slotted on the main mother board, the only direct pin-for-pin replacement is the 68010.

For faster processors such as the 68020, 68030 or 68040 then a new expansion board needs to be purchased and inserted via the trapdoor expansion slot underneath the Amiga.

Typical boards include:

68020 Accelerators

ICD ADSpeed
68020 14MHz Accelerator.

Viper 520CD
Has a 33Mhz 68020 EC processor with room for 33Mhz FPU, space for a 2.5" IDE hard disk, 2 x 40 pin CD ROM socket and 8MB of RAM on board. Includes 3.0 ROM and software.

Solid State Leisure A5000
Includes a 16.67 68020 processor which uses 256x4 DRAMs (total of 4 1MB chips ie 5MB). Doesn't autoconfig or use DMA.

Supra 28 Turbo
Uses a 28MHz 68HC000 processor with high speed static RAM cache. Fits on the side expansion slot. Also, compatible with A2000.

68030 Accelerators

GVP A530
Includes a 40MHz 68EC030 processor with room for FPU and uses GVP SIMMS. Includes a on/off switch with turns off the 030/RAM and HD for backward compatibility. Up to 16/32 MB RAM.

Derringer 030
Uses a 25MHZ 68030 processor with optional 25/50Mhz FPU and upto 32MB of RAM using one 32 bit SIMM.

68040 Accelerators

Progressive Peripherals A500/040
This includes a 25Mhz 68040 processor with 4 or 8 MB 32-bit ZIP Fast RAM. Requires KS/WB 2.0 to work.

Upgrading the Kickstart and Workbench

The Amiga A500 can be upgraded to latest Kickstart and Workbench as follows:

From	To	Requirements
Workbench 1.3	Workbench 2.04 / 2.1	Kickstart 2
Workbench 1.3	Workbench 3.1	Kickstart 3.1
Workbench 1.3	Workbench 3.5 / 3.9	Kickstart 3.1, 6M RAM, 68020 processor, Hard Disk, CD ROM

In most cases a new Kickstart ROM Chip set needs to be installed.
You can also replace the old OCS chips with ECS (Enhanced Chip Set) but you cannot install AGA Chip Set.

Backward Compatibility

Compatibility with older software can be done in two ways: Hardware and Software.

Hardware:
a) Install a Kickstart ROM Switcher to allow you to switch between Kickstart 1.3 and later Kickstart

Software:
a) Use a SoftKicker program to load old Kickstart into memory e.g Skick
b) Use Disk to Hard disk programs such as JST and WHDLoad to copy the games to hard disk and emulate the older system.
c) Use CPU command to turn off features such as Caches on processors such as the 68020 or later.
d) Use the Kickstart Startup screen (press both mouse buttons at startup) to select which Chipset to use.