WB1_3

COLLABORATORS			
	TITLE : WB1_3		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 10, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

WB1_3

1.1 main

A Guide to Workbench 1.3 on the A500/A1500/A2000.

By PJ Hutchison © 22/4/00 v3.0

Contents

1.2 1. The Menus

Goto Contents

1.3 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.

1.4 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.

1.5 1.3 Special Menu

1.3.1 Cleanup

This option rearrages 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.

1.6 2. The Mouse

On the Amiga mouse there are two buttons. The Left Mouse Button (LMB) and the Right Mouse Button (RMB).

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.

1.7 3 WB Programs

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

3.1 Workbench Disk

3.2 Extras Disk

Goto Contents

1.8 3.1 Workbench Disk

3.1.1 Utilities Drawer

```
     Calculator     - Program to work out arithmetic calculations
     Clock          - Displays a analogic or digital clock showing
                      the present time and/or date
     ClockPtr       - Turns the pointer from an arrow to a clock!
     Cmd            - Redirects output to the printer to a file
     GraphicDump    - Prints out graphics and pictures
     InstallPrinter - Installs Printer driver in devs/printers
     More           - Program to display text files
     Notepad        - A simple text editor
     Printfiles     - Prints text files
     Say            - Speaks any text you type in
```

3.1.2 System Drawer

```
     CLI            - Opens a Command Line Interface console
     Diskcopy       - Program to copy disks (used by Duplicate)
     Format         - Formats a disk (used by Initialise)
     Fastmemfirst   - Tells Amiga to load programs into Fast memory
     SetMap         - Selects the keyboard map from devs/keymaps
     InitPrinter    - Initalises printer (same as switching it off and on).
     NoFastMem      - Disables all fast memory
     Mergemem       - Adds other memory to be used by the Amiga
     Fixfonts       - Updates Fonts files if changes have been made to
                      any files in Fonts drawer
```

3.1.3 Prefs Drawer

```
     Preferences    - Changes the settings used by the Amiga such as
                      the pointer, printer, serial port etc.
     CopyPrefs      - Makes a copy of prefs to be copied to another
                      disk
     Pointer        - Displays the pointer prefs screen.
     Printer        - Displays the printer prefs screen.
     Serial         - Displays the serial prefs screen.
```

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.
```

1.9 3.2 Extras Disk

```
                AmigaBASIC     - BASIC Programming Lanaguage interpretor

  Complete list of
                Amiga Basic Commands
                 is here.

3.2.1 Tools Drawer

     MEmacs    - Text editor
     Fed       - Font editor
     Freemap   - Memory usage display
     PrefMon   - Performance monitor
     IconEd    - Icon editor
     Palette   - Changes Workbench colour scheme
     KeyToy2000 - Displays keyboard characters
     IconMerge - Creates dual-image icons

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
```

1.10 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

                4.2 Error messages

                4.3 Volume/Drawer/File names

                4.4 Volumes,Drawers and Assigns
```

1.11 4.1 Command format

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

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

1.12 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	Stram 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
```

1.13 4.3 Volume/Drawer/File names

Names can be upto 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.

1.14 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 colin ':' e.g. DIR DataDisk:

A drawer or directory is used to organise files together
and they are usually used in conjunction which a slash to
seperate 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
```

1.15 4.5 Devices

```
A device is a special name given to a particular peripheral
or an AmigaDOS 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)
```

1.16 4.6 The Commands

```
Addbuffers     - Add memory to floppy disk buffers to improve speed
Ask            - Asks a question in a script (reply y/n). Use IF WARN
                 to test for a yes.
Assign         - Assign names to directory paths.
Avail          - Memory availablility
Binddrivers    - Load device drivers
Break          - Stop a program (see Status)
CD             - Change directory
ChangeTaskPri  - Changes program's execution priority
Copy           - Copy files
Date           - Display/change date or time
Delete         - Delete files
Dir            - List contents of a directory
Diskchange     - Tell Amigados that a disk has been swapped in a
                 non-autosensing disk drive
```

```
Diskdoctor     - Recover files from a disk
Echo           - Display text in a script
Ed             - Edit text files (screen editor)
Edit           - Edit text files (line editor)
Else           - Used with IF, if not true do these commands
Endcli         - Close CLI console window
Endif          - End a IF..ELSE..ENDIF block
Endskip        - Stops a SKIP branch (for debugging)
Eval           - Evaluate a simple arithmetic expression
Execute        - Execute a script or batch file
Failat         - Change failure limit for scripts
                 (0 - Ok, 5 - Warn, 10 - Error, 20 - Fail)
Fault          - Display error message from a code
FF             - Enable Fast fonts
Filenote       - Attach a comment to a file (see LIST)
Getenv         - Return contents of an environment variable
Iconx          - 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.
If             - If condition is true execute following commands until
                 an ELSE or ENDIF occurs.
Info           - Displays information about disks
Install        - 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.
Join           - Join two or files together.
Lab            - State a Label to Skip to in a script
List           - List file details in a directory
Loadwb         - Display workbench screen
Lock           - Lock a device from writing
Makedir        - Create a new directory.
Mount          - Loads and mount a device
Newcli         - Open a new CLI console (CON) window
NewShell       - Open a new Shell console (NEWCON) window
Sort           - Sort contents of a file
Stack          - Change/display amount of stack space
Status         - Display status of running programs
Type           - Display contents of text file on screen
Path           - Set or change the program search path
Prompt         - Change the prompt text
Protect        - Change the 'rwedsp' flags of a file. (r=read,w=write,
                 e=execute,d=delete,s=script,p=pure)
Quit           - Quit a script
Relabel        - Change name of a disk
RemRAD         - Remove RAD device from memory
Resident       - Stores AmigaDOS command in memory (quicker then reloading
                 from disk each time it is executed). To ensure a
                 command can be made resident check the pure flag
                 (see List).
Run            - Execute a program in the background (multi-task)
Search         - Search for a string in a file or a file on a disk
Setclock       - Load or set the date and time from battery-backed clock
Setdate        - Set date of a file (see List)
Setenv         - Change the contents of a environment variable
Setpatch       - Patch Operating System functions and remove bugs
Skip           - Jump to a label in a script (see Lab)
Version        - Display workbench or library version information
```

```
Wait           - Wait for a specified period of time
Which          - Find where a file is in command path (see Path)
Why            - Display reason why a previous command failed
```

1.17 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 ↩
.

1.18 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.

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 its a text file, change it to 'More' (Workbench's Text Viewer).

1.19 Improving 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).

Of course to keep really upto date is to upgrade to Workbench 3.1 and there are many advantages to do so:

a) Be able to run latest programs - commercial and PD.

b) Includes CrossDOS to read PC disks.

c) Includes Datatypes to access many file types esp. from other OS'
   and access to the Internet.

d) Compatibility is very high and there is way to degrade to 1.3 for
   older software (no AGA or fast processors in the way either).
   Use some of the Kicker software for this feature eg. SKick, Relokick.

e) Arexx included for program interaction and batch procesiing.

1.20 The Author

This guide has been written for those people would like to make their Workbench more up to date and more productive for their use. I hope some of the information will be useful. Almost all the programs described can be readily installed and used from the many utility collections, the best place is the Aminet archives either via the Internet or on CD. There are some CDs with Workbench Enhancements included as one collection and are worth looking at.

How to contact me:

Peter Hutchison

E-Mail: P.J.Hutchison@hud.ac.uk or pjhutch@blizzard.u-net.com
Web: http://www.blizzard.u-net.com/index.html
Fidonet: 2:250/366.24

Other Guides to look out for:

The Printer Guide
Expanding the Amiga Guide
PC Task Guide
Upgrading Workbench
Foozle Guide
Hard Disk Guide

See Aminet/docs/help or hyper for the above.