

# App. G: Command Index

---

This Command Index may well be one of the most useful sections of the User Guide for genuine AOZ programmers. For on-line explanations and examples of all keywords, the [Help] facility is invaluable, but this Appendix is the only facility which provides a complete overview of all of the AMOS Professional commands. Experienced users can scan this Index to embrace everything that the system has to offer.

The Command Index includes every instruction, function, structure and reserved variable in alphabetical order, along with a synopsis of usage. Every embedded Menu command, Interface instruction and AMAL keyword is also included.

The page references refer to the main explanation of each keyword that can be found in this User Guide. Page references are shown using the following protocol: Section.Chapter.Page. For example, a keyword which is referenced as 5.3.01 means that full details can be found in Section 5, Chapter 3, Page 01 of this User Guide.

For associated items, cross-references and general topics, please refer to the Main Index at the end of this User Guide.

| Command | Type | Description | Chapter |

---|---|---|---

|ABS | Function | give an absolute value | 5.3.04 |

|ACOS | Function | give arc cosine | 5.3.09 |

|ADD | Instruction | perform fast integer addition | 5.3.02 |

AL Interface | Instruction | display an active list window | 9.3.10

AMAL | Instruction | call an AMAL program | 7.6.11

AMAL FREEZE | Instruction | suspend AMAL programs | 7.6.18

AMAL OFF | Instruction | stop all AMAL programs | 7.6.17

AMAL ON | Instruction | activate all AMAL programs | 7.6.17

AMALERR | Function | give the position of an AMAL error | 7.6.20

AMOS HERE | Function | report if AMOS Pro is at front of display | 11.4.02

AMOS LOCK | Instruction | disable [Amiga]+[A] toggle facility | 11.4.01

AMOS TO BACK | Instruction | hide AMOS Professional and reveal Workbench | 11.4.01

AMOS TO FRONT | Instruction | hide Workbench and reveal AMOS Professional | 11.4.01

AMOS UNLOCK | Instruction | re-activate AMOS Professional/Workbench toggle | 11.4.02

AMPLAY | Instruction | control animation produced by PLay | 7.6.18

AM REG | Reserved Variable | give the value of an AMAL register | 7.6.18

AND | Structure | qualify a condition | 5.4.03

ANIM | Instruction | animate an Object | 7.6.24

ANIM FREEZE | Instruction | freeze an animation | 7.6.25

ANIM OFF | Instruction | toggle animations off | 7.6.24

ANIM ON | Instruction | toggle animations on | 7.6.24

Anim AMAL | Instruction | animate an Object | 7.6.03

APPEAR | Instruction | fade between two screens | 6.3.01

APPEND | Instruction | add data to an existing file | 10.2.11

AR Interface | Instruction | read an element from an array | 9.3.09

AREG | Reserved Variable | pass values to and from an address register | 14.A.15

AREXX | Function | check for a message from an AREXX program | 10.6.03

AREXX ANSWER | Instruction | answer a message from an AREXX program | 10.6.04  
AREXX CLOSE | Instruction | close a communications port | 10.6.02  
AREXX EXIST | Function | check the availability of a communications port | 10.6.02  
AREXX OPEN | Instruction | open an AREXX communications port | 10.6.02  
AREXX\$ | Function | get a message from an AREXX program | 10.6.03  
AREXX WAIT | Instruction | wait for a message from an AREXX program | 10.6.03  
ARRAY | Function | load the address of an array into a program | 9.3.09  
AS | Instruction | please see RESERVE | 5.9.02  
AS Interface | Instruction | return the size of an array | 9.3.10  
ASC | Function | give the Ascii code of a character | 5.2.05  
ASIN | Function | give arc sine | 5.3.09  
ASK EDITOR | Instruction | return params from Editor to an accessory program | 13.1.02  
ASSIGN | Instruction | assign a name to a path or device | 10.2.06  
AT | Function | return a string to position the text cursor | 5.6.06  
ATAN | Function | give arc tangent | 5.3.09  
AUTO VIEW OFF | Instruction | toggle viewing mode off | 6.1.03  
AUTO VIEW ON | Instruction | toggle viewing mode on | 6.1.03  
AUTOBACK | Instruction | set the graphics mode on double buffered screen | 7.3.06  
AUtotest AMAL | Instruction | activate AMAL Autotest system | 7.6.06  
BA Interface | Instruction | set coordinate base for dialogue box | 9.1.07  
BANK SHRINK | Instruction | reduce the size of a memory bank | 5.9.07  
BANK SWAP | Instruction | swap over two memory banks | 5.9.07  
BANK TO MENU | Instruction | restore menu definition saved in menu bank | 6.5.06  
BAR | Instruction | draw a filled rectangle | 6.4.08  
BAr | Embedded Menu Command | draw a bar | 6.5.15  
BC AMAL | Function | check for Bob collision | 7.6.08  
BC Interface | Instruction | change the setting of any active button | 9.1.13  
BCHG | Instruction | toggle a bit | 14.A.10  
BCLR | Instruction | clear a bit | 14.A.11  
BELL | Instruction | generate a pure audio tone | 8.1.01  
BGRAB | Instruction | grab a memory bank from previous program | 5.9.10  
BIN\$ | Function | convert a decimal value to binary number | 14.A.02  
BLENGTH | Function | give the length of a memory bank from previous program | 5.9.09  
BLOAD | Instruction | load block of binary data into a bank or an address | 5.9.05  
BO Interface | Instruction | draw a box from Resource Bank image components | 9.4.02  
BOB | Embedded Menu Command | draw a Bob | 6.5.14  
BOB | Instruction | display a Bob on screen | 7.2.01  
BOB CLEAR | Instruction | clear all Bobs from the screen | 7.3.05  
BOB COL | Function | test for collision between Bobs | 7.4.03  
BOB DRAW | Instruction | re-draw all Bobs on screen | 7.3.05  
BOB OFF | Instruction | remove a Bob from display | 7.2.03  
BOB UPDATE | Instruction | move several Bobs simultaneously | 7.3.03  
BOB UPDATE OFF | Instruction | turn off automatic Bob update system | 7.3.03  
BOB UPDATE ON | Instruction | turn on automatic Bob update system | 7.3.03  
BOBSPRITE COL | Function | test for collision between Bob and Sprites | 7.4.04  
BOOM | Instruction | generate explosive sound effect | 8.1.01  
BORDER | Instruction | change window border | 5.7.02  
BORDER\$ | Function | create a border around text | 5.6.11  
BOX | Instruction | draw a rectangular outline | 6.4.03  
BP Interface | Function | return the setting inside a button definition | 9.1.12  
BQ Interface | Instruction | trigger an exit button | 9.1.11  
BR Interface | Instruction | change the setting of a button | 9.1.13

BREAK OFF | Instruction | toggle off program break keys | 5.1.08  
BREAK ON | Instruction | toggle on program break keys | 5.1.08  
BSAVE | Instruction | save unformatted memory bank | 5.9.05  
BSEND | Instruction | send a memory bank to previous program | 5.9.10  
BSET | Instruction | set a bit to 1 | 14.A.10  
BSTART | Function | give address of a memory bank from a previous program | 5.9.10  
BTST | Function | test a bit | 14.A.10  
BU Interface | Instruction | define an Interface button | 9.1.09  
BX Interface | Function | get the x-coordinate base location | 9.2.01  
BY Interface | Function | get the y-coordinate base location | 9.2.01  
C AMAL | Function | give collision status of an Object | 7.6.09  
CALL | Instruction | execute a machine code program | 14.A.14  
CALL EDITOR | Instruction | send | Instruction | s to Editor from an accessory program | 13.1.01  
CALL Interface | Instruction | call a machine code extension | 9.2.10  
CDOWN | Instruction | move the text cursor down | 5.6.08  
CDOWN\$ | Function | return control character to move text cursor down | 5.6.08  
CENTRE | Instruction | print text at centre of current line | 5.6.07  
CHANAN | Function | test a channel for an active animation | 7.6.19  
CHANGE MOUSE | Instruction | change the shape of the mouse pointer | 5.8.03  
CHANMV | Function | test channel for an active Object | 7.6.19  
CHANNEL | Instruction | assign an Object to an AMAL channel | 7.6.12  
CHIP FREE | Function | give remaining Chip memory | 3.1.05  
CHOICE | Function | read a menu | 6.5.02  
CHR\$ | Function | give a character with a given Ascii code | 5.2.05  
CIRCLE | Instruction | draw a circular outline | 6.4.03  
CLEAR KEY | Instruction | re-set the keyboard buffer | 10.1.03  
CLEFT | Instruction | move text cursor one character to the left | 5.6.08  
CLEFT\$ | Function | move cursor 1 character to the left | 5.6.09  
CLINE | Instruction | clear text on the current cursor line | 5.6.09  
CLIP | Instruction | restrict drawing to a limited screen area | 6.4.04  
CLOSE | Instruction | close a file | 10.2.12  
CLOSE EDITOR | Instruction | close the AMOS Professional editor | 13.1.07  
CLOSE WORKBENCH | Instruction | close the Workbench | 13.1.07  
CLS | Instruction | clear the current screen | 6.1.07  
CLW | Instruction | clear the current window | 5.7.04  
CMOVE | Instruction | move the text cursor | 5.6.06  
CMOVE\$ | Function | return control string to position text cursor | 5.6.06  
COL | Function | test status of Object after collision detect routine | 7.4.04  
COLOUR | Function | read the colour assignment | 6.4.05  
COLOUR | Instruction | assign colour to an index | 6.4.06  
COLOUR BACK | Instruction | assign colour to screen background | 6.4.06  
COMMAND LINE\$ | Reserved Variable | transfer parameters between programs | 10.2.09  
COP LOGIC | Function | give address of logical copper list | 14.F.04  
COP MOVE | Instruction | write a Move Instruction to the copper list | 14.F.03  
COP MOVE L | Instruction | write lone Move Instruction to the copper list | 14.F.03  
COP RESET | Instruction | re-set copper list pointer | 14.F.04  
COP SWAP | Instruction | swap over the logical and physical copper lists | 14.F.04  
COP WAIT | Instruction | insert a Wait Instruction into copper list | 14.F.03  
COPPER OFF | Instruction | turn off standard copper list | 14.F.01  
COPPER ON | Instruction | re-start automatic copper generation | 14.F.03  
COPY | Instruction | copy a memory block | 14.A.05  
COS | Function | give cosine of an angle | 5.3.08

CRIGHT | Instruction | move the text cursor one character to the right| 5.6.08  
CRIGHT\$ |Function| move cursor one character right |5.6.09  
CUP | Instruction | move the text cursor up one line| 5.6.08  
CUP\$ |Function| return control character to move cursor up one line| 5.6.09  
CURS OFF | Instruction | toggle the text cursor off |5.6.11  
CURS ON | Instruction | toggle the text cursor on |5.6.11  
CURS PEN | Instruction | select colour of the text cursor |5.6.10  
CX Interface |Function| centre text in the display |9.2.05  
DATA |Structure| place a list of data items in a program |5.4.12  
DEC | Instruction | decrement an integer variable by one unit |5.3.02  
DEEK |Function| read two bytes from an even address |14.A.04  
DEF FN |Structure| create a user-defined function |5.1.06  
DEF SCROLL | Instruction | define a scrolling screen zone| 6.2.02  
DEFAULT | Instruction | re-set to the default screen |6.1.03  
DEFAULT PALETTE | Instruction | define standard palette |6.1.09  
DEGREE | Instruction | use degrees |5.3.08  
DEL BLOCK | Instruction | delete a screen block| 7.7.04  
DEL BOB | Instruction | delete an image from the Object bank| 7.2.09  
DEL CBLOCK | Instruction | delete compacted screen block |7.7.05  
DEL ICON | Instruction | delete image from the Icon bank |7.7.02  
DEL SPRITE | Instruction | delete an image from the Object bank |7.1.05  
DEL WAVE | Instruction | delete an audio wave |8.1.07  
DEV ABORT | Instruction | abort an IO operation| 11.5.08  
DEV BASE |Function| get base address of an IO structure |11.5.08  
DEV CHECK |Function| check status of a device with CheckIO |11.5.08  
DEV CLOSE | Instruction | close one or more devices |11.5.07  
DEV DO | Instruction | call a command using DoIO |11.5.08  
DEV FIRST\$ |Function| get the first device from the current device list| 11.5.06  
DEV NEXT\$ |Function| get the next device in the current search path| 11.5.07  
DEV OPEN | Instruction | open a device |11.5.07  
DEV SEND | Instruction | call a command using SendIO| 11.5.08  
DFREE |Function| report amount of free space on disc |10.2.10  
DI Interface | Instruction | create a numeric editing zone| 9.3.07  
DIALOG |Function| return the status of an open dialogue box |9.3.03  
DIALOG BOX |Function| display dialogue box on the screen |9.1.06  
DIALOG CLOSE | Instruction | close one or more dialogue channels| 9.3.02  
DIALOG CLR | Instruction | clear a dialogue box |9.3.15  
DIALOG FREEZE | Instruction | stop dialogue channel input |9.3.15  
DIALOG OPEN | Instruction | open a channel to an Interface program |9.3.01  
DIALOG RUN |Function| run a dialogue box from an open channel |9.3.02  
DIALOG UNFREEZE | Instruction | re-activate a frozen dialogue channel| 9.3.15  
DIALOG UPDATE | Instruction | update a dialogue zone| 9.3.14  
DIM | Instruction | dimension an array |5.1.04  
DIR | Instruction | print directory of the current disc |10.2.02  
DIR FIRST\$ |Function| get the first file that satisfies a path name| 10.2.07  
DIR NEXT\$ |Function| get the next file that satisfies path name |10.2.07  
DIR\$ |Reserved Variable| change the current directory |10.2.05  
DIR/W | Instruction | print directory in two columns |10.2.02  
DIRECT | Instruction | return to Direct Mode| 5.1.08  
Direct AMAL | Instruction | change resumption point of main program| 7.6.06  
DISC INFO\$ |Function| report free space of named file or directory |10.2.10  
DISPLAY HEIGHT |Function| give the maximum available screen height |14.C.01

DO |Structure| mark the beginning of a loop |5.4.06

DOKE | Instruction | change two-byte word at an even address| 14.A.04

DOSCALL |Function| execute function from the DOS library |11.5.02

DOUBLE BUFFER | Instruction | activate the double buffering system| 7.2.06

DRAW | Instruction | draw a line |6.4.02

DRAW TO | Instruction | draw a line from the last coordinates| 6.4.02

DREG |Reserved Variable| pass a value into 68000 data register |14.A.15

DUAL PLAYFIELD | Instruction | combine two screens |6.1.06

DUAL PRIORITY | Instruction | reverse the order of dual playfield screens| 6.1.07

ED Interface | Instruction | create a text edit zone |9.3.05

[DIALOG |Function| find an error in an Interface program |9.3.03

EDIT | Instruction | return to the Edit Screen |5.1.08

ELLIPSE | Instruction | draw an elliptical outline |6.4.04

ELipse |Embedded Menu Command| draw an ellipse |6.5.16

ELSE |Structure| qualify a condition |5.4.04

ELSE IF |Structure| qualify a condition| 5.4.05

END | Instruction | stop the current program| 5.1.07

End AMAL | Instruction | terminate an AMAL program |7.6.06

END IF |Structure| terminate a structured test |5.4.04

END PROC |Structure| end a procedure |5.5.01

EOF |Function| test for end of file |10.2.13

EQU |Function| get an equate used by Amiga system library| 11.5.04

ERASE | Instruction | clear a single memory bank |5.9.06

ERASE ALL | Instruction | clear all current memory banks |5.9.06

ERASE TEMP | Instruction | clear temporary memory banks |5.9.07

ERRN |Function| give error code number of error |12.2.03

ERROR | Instruction | generate an error |12.2.03

ERR\$ |Function| return an error message string| 12.2.04

ERRTRAP |Function| return an error code number after a Trap command| 12.2.04

EVERY | Instruction | call a sub-routine or a procedure at regular intervals |5.4.11

EVERY OFF | Instruction | toggle regular calls off |5.4.12

EVERY ON | Instruction | toggle regular calls on |5.4.12

EX Interface | Instruction | exit from Interface and return to main program |9.1.03

EXEC | Instruction | send a CLI command to a device |11.4.04

EXECALL |Function| call EXEC library |11.5.02

EXIST |Function| check if specified file exists |10.2.06

EXIT |Structure| break out of a loop |7.6.16

eXit AMAL | Instruction | leave Autotest and return to main program |7.6.06

EXIT IF |Structure| break out of a loop depending on a test |5.4.07

EXP |Function| calculate an exponential number| 5.3.06

FADE | Instruction | blend colours to new values |6.3.01

FALSE |Function| hold a value of zero if a condition is false| 5.4.06

FAST FREE |Function| return amount of Fast memory in bytes |3.1.05

FIELD | Instruction | define a record structure |10.2.14

FILL | Instruction | fill memory block with the contents of a variable |14.A.05

FIRE |Function| test the status of joystick fire-button |5.8.02

FIX | Instruction | fix the precision of floating point |5.3.05

FLASH | Instruction | set flashing colour sequence |6.3.03

FLASH OFF | Instruction | turn off the flashing colour sequence| 6.3.03

FLIP\$ |Function| invert a string |5.2.04

FN |Structure| call a user-defined function |5.1.06

FONT\$ |Function| return details of available fonts |11.1.02

FOR | Structure | mark the beginning of a loop | 5.4.09  
For AMAL | Structure | mark beginning of a loop | 7.6.05  
FRAME LENGTH | Function | give frame length in bytes | 7.5.04  
FRAME LOAD | Function | load frames into memory | 7.5.04  
FRAME PARAM | Function | give parameter after playing a frame | 7.5.06  
FRAME PLAY | Function | play frames on screen | 7.5.05  
FRAME SKIP | Function | skip past an animation frame | 7.5.06  
FREE | Function | give free memory available in variable area | 5.1.05  
FREEZE | Instruction | freeze the display | 7.5.07  
FSEL\$ | Function | select a file | 10.2.07  
GB Interface | Instruction | draw a graphic filled box | 9.2.02  
GE Interface | Instruction | draw an ellipse or circle | 9.2.04  
GET | Instruction | read a record from random access file | 10.2.15  
GET BLOCK | Instruction | grab a screen block into memory | 7.7.03  
GET BOB | Instruction | grab an image from part of the screen | 7.2.07  
GET BOB PALETTE | Instruction | load image colours to current screen | 7.2.04  
GET CBLOCK | Instruction | save and compact a screen block | 7.7.04  
GET DISC FONTS | Instruction | create a list of available fonts from current disc | 11.1.02  
GET FONTS | Instruction | create a list of available fonts from Rom and disc | 11.1.01  
GET ICON | Instruction | create an icon | 7.7.01  
GET ICON PALETTE | Instruction | load icon colours into current screen | 7.7.01  
GET PALETTE | Instruction | copy palette from a screen | 6.1.10  
GET ROM FONTS | Instruction | create a list of available ROM fonts | 11.1.02  
GET SPRITE | Instruction | grab screen image into Object bank | 7.1.07  
GET SPRITE PALETTE | Instruction | grab Sprite colours into current screen | 7.1.07  
GFXCALL | Function | call Graphics library | 11.5.03  
GL Interface | Instruction | draw a line on screen | 9.2.04  
GLOBAL | Structure | declare list of global variables | 5.5.06  
GOSUB | Structure | jump to a sub-routine | 5.4.02  
GOTO | Structure | jump to defined position in a program | 5.4.01  
GR LOCATE | Instruction | position the graphics cursor | 6.4.01  
GR WRITING | Instruction | set the graphic writing mode | 6.4.10  
GS Interface | Instruction | draw a graphic hollow rectangle | 9.2.03  
HARDCOL | Function | return collision status after a Set Hardcol Instruction | 7.4.05  
HCOS | Function | give hyperbolic cosine | 5.3.10  
HEX\$ | Function | convert decimal value to hexadecimal number | 14.A.02  
HIDE | Instruction | remove the mouse pointer from the screen | 5.8.03  
HIDE ON | Instruction | keep mouse pointer hidden from the screen | 5.8.03  
HIRES | Function | set screen mode to 640 pixels wide | 6.1.02  
HOME | Instruction | force the text cursor home | 5.6.06  
HOT SPOT | Instruction | set reference point for all coordinate calculations | 7.1.11  
HREV | Function | flip an image horizontally | 7.2.10  
HREV BLOCK | Instruction | flip a block horizontally | 7.7.04  
HS Interface | Instruction | create an animated horizontal slider bar | 9.3.07  
HSCROLL | Instruction | scroll text horizontally | 5.6.12  
HSIN | Function | give hyperbolic sine | 5.3.09  
HSLIDER | Instruction | draw a horizontal slider bar | 5.7.05  
HT Interface | Instruction | open an interactive text window | 9.3.15  
HTAN | Function | give hyperbolic tangent | 5.3.10  
HUNT | Function | find a string of characters in memory | 14.A.06  
HZONE | Function | give screen zone under hardware coordinates | 7.4.07  
I BOB | Function | get image number used by a Bob | 7.2.04

I SPRITE | Function | get image number used by a Sprite | 7.1.09  
Icon | Embedded Menu Command | draw an icon | 6.5.14  
ICON BASE | Function | get icon base | 5.9.11  
IF | Structure | choose between alternative statements | 5.4.03  
IF Interface | Structure | mark the start of a conditional test | 9.2.07  
If AMAL | Structure | perform a test | 7.6.05  
IFF ANIM | Instruction | play an animation file | 7.5.03  
IL Interface | Instruction | display an inactive list window | 9.3.11  
IN Interface | Instruction | set the current drawing colour | 9.2.03  
INC | Instruction | increment an integer variable by one unit | 5.3.02  
INCLUDE | Instruction | specify a file for inclusion when testing a program | 10.2.16  
Ink | Embedded Menu Command | set colour | 6.5.14  
INK | Instruction | set drawing colour | 6.4.05  
INKEY\$ | Function | check for a key press | 10.1.01  
INPUT | Instruction | load a value into a variable | 10.1.04  
INPUT\$ | Function | anticipate characters to be input into a string | 10.1.04  
INPUT# | Structure | input variables from a file or device | 10.2.12  
INS BOB | Instruction | insert blank Bob image into Object bank | 7.2.10  
INS ICON | Instruction | insert a blank icon image into the Icon bank | 7.7.02  
INS SPRITE | Instruction | insert a blank Sprite image into the Object bank | 7.1.05  
INSTR | Function | search for one string inside another string | 5.2.02  
INT | Function | convert a floating point number into an integer | 5.3.04  
INTCALL | Function | call Intuition library | 11.5.03  
INVERSE OFF | Instruction | toggle inverse text off | 5.6.03  
INVERSE ON | Instruction | toggle inverse text on | 5.6.03  
J0 AMAL | Function | give status of right joystick | 7.6.09  
J1 AMAL | Function | give status of left joystick | 7.6.09  
JDOWN | Function | test joystick for downward movement | 5.8.02  
JLEFT | Function | test joystick for left movement | 5.8.01  
JOY | Function | read status of the joystick | 5.8.01  
JRIGHT | Function | test joystick for right movement | 5.8.02  
JS Interface | Instruction | call an Interface sub-routine | 9.2.07  
JP Interface | Instruction | jump to an Interface program label | 9.2.06  
Jump AMAL | Instruction | jump to a label | 7.6.03  
JUP | Function | test joystick for upward movement | 5.8.02  
K1 AMAL | Function | give status of left mouse key | 7.6.09  
K2 AMAL | Function | give status of right mouse key | 7.6.09  
KEY SHIFT | Function | test status of shift keys | 10.1.03  
KEY SPEED | Instruction | set key repeat speed | 10.1.06  
KEY STATE | Function | test for a specific key state | 10.1.02  
KEY\$ | Reserved Variable | define a keyboard macro | 10.1.06  
KILL | Instruction | erase a file from the current disc | 10.2.10  
KILL EDITOR | Instruction | remove the AMOS Professional Editor from memory | 14.B.01  
KY Interface | Instruction | set a keyboard short-cut | 9.1.14  
LA Interface | Instruction | create a simple label | 9.2.06  
LACED | Function | give value linked to screen resolution | 6.1.13  
LDIR | Instruction | output directory of the current disc to a printer | 10.2.04  
LDIR/W | Instruction | output directory of disc in 2 columns to printer | 10.2.04  
LED OFF | Instruction | toggle audio filter off | 8.1.08  
LED ON | Instruction | toggle audio filter on | 8.1.08  
LEEK | Function | read four bytes from an even address | 14.A.04  
LEFT\$ | Function | give the leftmost characters of a string | 5.2.01

LEN | Function | give the length of a string | 5.2.05  
LENGTH | Function | give the length of a memory bank | 5.9.08  
Let AMAL | Instruction | assign a value to a register | 7.6.04  
LIB BASE System | Function | get the base address of system library | 11.5.02  
LIB CALL System | Function | call a function from a system library | 11.5.01  
LIB CLOSE System | Function | close one or all currently open system libraries | 11.5.01  
LIB OPEN | Instruction | open a system library for use | 11.5.01  
LIMIT BOB | Instruction | limit Bob to part of the screen | 7.2.06  
LIMIT MOUSE | Instruction | limit the mouse pointer to part of the screen | 5.8.06  
Line Interface | Instruction | draw a line of Resource Bank image components | 9.4.02  
Line | Embedded Menu | Command draw a line | 6.5.15  
LINE INPUT | Instruction | input a list of variables separated by [Return] | 10.1.05  
LINE INPUT# | Structure | input list of variables not separated by a comma | 10.2.12  
LIST BANK | Instruction | list all current banks in memory | 5.9.08  
LN | Function | give natural logarithm | 5.3.06  
LOAD IFF | Instruction | load an IFF screen from disc | 6.1.11  
LOAD | Instruction | load one or more banks into memory | 5.9.04  
LOcate | Embedded Menu Command | move graphics cursor | 6.5.13  
LOCATE | Instruction | position the text cursor | 5.6.05  
LOF | Function | give the length of an open file | 10.2.13  
LOG | Function | give logarithm | 5.3.06  
LOGBASE | Function | give the address of logical screen bit-plane | 6.2.04  
LOGIC | Function | give number of the logical screen | 6.2.04  
LOKE | Instruction | change a four-byte word at an even address | 14.A.04  
LOOP | Structure | mark the end of a loop | 5.4.06  
LOWER\$ | Function | convert a string of text to lower case | 5.2.03  
LOWRES | Function | set screen resolution to 320 pixels wide | 6.1.02  
LPRINT | Instruction | output a list of variables to a printer | 5.6.14  
LVO | Function | get the Library Vector Offset | 11.5.04  
MAKE ICON MASK | Instruction | set colour zero to transparent | 7.7.03  
MAKE MASK | Instruction | mask an image for collision detection | 7.4.02  
MASK IFF | Instruction | mask IFF picture data | 7.5.07  
MATCH | Function | search an array for a value | 5.2.06  
MAX | Function | give the maximum of two values | 5.3.03  
ME Interface | Function | return a message from the Resource Bank | 9.4.01  
MED CONT | Instruction | continue a Med module | 8.3.04  
MED LOAD | Instruction | load a Med music module | 8.3.03  
MED MIDI ON | Instruction | access MIDI Instructions in a Med module | 8.3.04  
MED PLAY | Instruction | play a Med module | 8.3.04  
MED STOP | Instruction | stop the current Med module | 8.3.04  
MEMORIZE X | Instruction | save the x-coordinate of the text cursor | 5.6.09  
MEMORIZE Y | Instruction | save the y-coordinate of the text cursor | 5.6.09  
MENU ACTIVE | Instruction | activate a menu item | 6.5.08  
MENU BAR | Instruction | display menu items as a vertical bar | 6.5.07  
MENU BASE | Instruction | move the starting position of a menu | 6.5.11  
MENU CALC | Instruction | recalculate a menu | 6.5.06  
MENU CALLED | Instruction | re-draw a menu item continually | 6.5.17  
MENU DEL | Instruction | delete one or more menu items | 6.5.06  
MENU INACTIVE | Instruction | turn off a menu item | 6.5.08  
MENU ITEM MOVABLE | Instruction | move individual menu items | 6.5.09  
MENU ITEM STATIC | Instruction | fix menu items in a static position | 6.5.09  
MENU KEY | Instruction | assign a key to a menu item | 6.5.12



MENU LINE | Instruction | display menu options in a horizontal line |6.5.07  
MENU LINK | Instruction | link a list of menu items |6.5.10  
MENU MOUSE OFF | Instruction | toggle off menu under the mouse pointer| 6.5.11  
MENU MOUSE ON | Instruction | toggle on menu under the mouse pointer |6.5.11  
MENU MOVABLE | Instruction | activate automatic menu movement |6.5.09  
MENU OFF | Instruction | de-activate a menu |6.5.05  
MENU ON | Instruction | activate a menu |6.5.05  
MENU ONCE | Instruction | turn off automatic menu re-drawing| 6.5.18  
MENU SEPARATE | Instruction | separate a list of menu items |6.5.10  
MENU STATIC | Instruction | fix menu in a static position |6.5.09  
MENU TLINE | Instruction | display menu as a total line |6.5.07  
MENU TO BANK | Instruction | save menu definitions to memory bank| 6.5.06  
MENU\$ | Reserved Variable| define a menu title or an option |6.5.01  
MID\$ | Function| give characters from the middle of a string |5.2.02  
MIN | Function| give the minimum of two values |5.3.03  
MKDIR | Instruction | create a folder |10.2.08  
MONITOR | Instruction | call the AMOS Professional Monitor| 12.1.01  
MOUSE CLICK |Function| check for mouse button click |5.8.05  
MOUSE KEY |Function| read the status of mouse buttons| 5.8.05  
MOUSE SCREEN |Function| check which screen the mouse pointer is in| 5.8.06  
MOUSE ZONE |Function| check if mouse pointer is in a zone |7.4.07  
MOUTH HEIGHT |Function| give height of the narrator mouth |11.2.03  
MOUTH READ |Function| read position of the narrator voice |11.2.03  
MOUTH WIDTH |Function| give width of the narrator mouth |11.2.03  
MOVE FREEZE | Instruction | suspend Object movement |7.6.24  
MOVE OFF | Instruction | toggle movements off |7.6.23  
MOVE ON | Instruction | toggle movements on |7.6.23  
MOVE X | Instruction | move an Object horizontally |7.6.22  
MOVE Y | Instruction | move an Object vertically |7.6.23  
Move AMAL | Instruction | move an Object |7.6.02  
MOVON |Function| give movement status |7.6.23  
MULTI WAIT | Instruction | force a multi-task Wait Vbl| 11.4.01  
MUSIC | Instruction | play a piece of AMOS Professional music |8.3.01  
MUSIC OFF | Instruction | turn off all music |8.3.01  
MUSIC STOP | Instruction | stop a single passage of music |8.3.01  
MVOLUME | Instruction | set the volume of a piece of music |8.3.01  
NEXT |Structure |match a For in a loop |5.4.09  
Next AMAL| Structure| counter for a loop |7.6.05  
NO ICON MASK | Instruction | remove colour zero from Icon |7.7.03  
NO MASK | Instruction | remove colour zero mask from a Bob |7.2.04  
NOISE TO | Instruction | assign a noise wave to a sound channel| 8.1.07  
NOT |Structure| logical NOT operation |5.4.06  
NTSC |Function| identify an NTSC or PAL machine |14.C.03  
NW Interface | Instruction | specify a quick-release button |9.1.14  
ON |Structure |jump on recognising a variable |5.4.10  
On AMAL | Instruction | activate main program after Wait| 7.6.06  
ON BREAK PROC |Structure| jump to a procedure when break in program| 5.5.04  
ON ERROR| Structure| trap an error within a program |12.2.01  
ON ERROR PROC |Structure| trap an error using a procedure |12.2.02  
ON MENU DEL | Instruction | delete labels and procedures used by On Menu| 6.5.05  
ON MENU GOSUB | Instruction | automatic menu selection |6.5.04  
ON MENU GOTO | Instruction | automatic menu selection |6.5.05

ON MENU PROC | Instruction | automatic menu selection | 6.5.04  
ON MENU OFF | Instruction | toggle automatic menu selection off | 6.5.05  
ON MENU ON | Instruction | toggle automatic menu selection on | 6.5.05  
ON PROC | Structure | trigger a jump to a procedure | 5.5.03  
OPEN IN | Instruction | open a file for input | 10.2.11  
OPEN OUT | Instruction | open a file for output | 10.2.11  
OPEN PORT | Instruction | open a channel to an IO port | 10.3.06  
OPEN RANDOM | Instruction | open a channel to a random access file | 10.2.14  
OR | Structure | qualify a condition | 5.4.03  
OOutline | Embedded Menu | Command enclose a bar with an outline | 6.5.15  
PACK | Picture Compactor | Extension pack screen data | 6.2.06  
PAINT | Instruction | fill a screen area with colour | 6.4.07  
PALETTE | Instruction | set the current screen colours | 6.4.06  
PAPER | Instruction | set the colour of text background | 5.6.02  
PAPER\$ | Function | give control index number to set background colour | 5.6.03  
PARALLEL ABORT | Instruction | stop a parallel operation | 10.5.02  
PARALLEL BASE | Function | get the base address of the Parallel Port | 10.5.03  
PARALLEL CHECK | Function | report the availability of the Parallel port | 10.5.02  
PARALLEL CLOSE | Instruction | close the Parallel Port | 10.5.01  
PARALLEL ERROR | Function | check for an error in transmission via Parallel Port | 10.5.02  
PARALLEL INPUT\$ | Function | read a string from the Parallel Port | 10.5.02  
PARALLEL OPEN | Instruction | open the Parallel Port for reading and writing | 10.5.01  
PARALLEL OUT | Instruction | send data from memory to the Parallel Port | 10.5.02  
PARALLEL SEND | Instruction | send a string of characters to the Parallel Port | 10.5.01  
PARALLEL STATUS | Function | report the current status of the Parallel Port | 10.5.03  
PARAM | Function | return a parameter from a procedure | 5.5.08  
PARAM# | Function | return a real number variable from a procedure | 5.5.08  
PARAM\$ | Function | return a string variable from a procedure | 5.5.08  
PARENT | Instruction | negotiate a path through the current directory | 10.2.05  
PASTE BOB | Instruction | draw an image from the Object bank | 7.2.08  
PASTE ICON | Instruction | draw an Icon | 7.7.02  
PAttern | Embedded Menu Command | set a pattern | 6.5.15  
Pause AMAL | Instruction | pause an AMAL program | 7.6.06  
PEEK | Function | read a byte from an address | 14.A.03  
PEEK\$ | Function | read a string of characters from memory | 14.A.05  
PEN | Instruction | set the colour for text and drawing operations | 5.6.02  
PEN\$ | Function | give a control index number to set pen colour | 5.6.02  
PHYBASE | Function | give address of the current screen | 6.2.04  
PHYSIC | Function | give the physical screen number | 6.2.04  
PI# | Function | give a constant PI | 5.3.07  
PICTURE | Function | give mask data of an IFF image | 7.5.07  
PLAY | Instruction | play a voice | 8.1.04  
PLAY OFF | Instruction | stop playing a voice | 8.1.04  
Play AMAL | Instruction | create a movement path | 7.6.07  
PLOAD | Instruction | load machine code from memory | 14.A.14  
PLOT | Instruction | plot a single point | 6.4.01  
PO Interface | Instruction | print hollow outline text | 9.2.04  
POF | Reserved Variable | hold the current position of the file pointer | 10.2.13  
POINT | Function | give the colour of a point | 6.4.01  
POKE | Instruction | change a byte at an address | 14.A.03  
POKE\$ | Instruction | poke a string of characters into memory | 14.A.04  
POLYGON | Instruction | draw a filled polygon | 6.4.08

POLYLINE | Instruction | draw multiple lines | 6.4.03  
POP | Instruction | remove Return information | 5.4.02  
POP PROC | Structure | leave a procedure immediately | 5.5.03  
PORT | Function | test the readiness of a device | 10.3.07  
PR Interface | Instruction | print the contents of a variable to the screen | 9.1.04  
PRG STATE | Function | return status of how current program was run | 11.4.04  
PRG UNDER | Function | report the availability of program under current program | 11.4.03  
PRINT | Instruction | print items on screen | 5.6.01  
PRINT# | Structure | print variables to a file or device | 10.2.12  
PRINTER ABORT | Instruction | stop a printer operation | 10.3.05  
PRINTER BASE | Function | get the address of printer base | 10.3.06  
PRINTER CHECK | Function | give status of printer | 10.3.05  
PRINTER CLOSE | Instruction | close printer port | 10.3.01  
PRINTER DUMP | Instruction | print the contents of a screen | 10.3.03  
PRINTER ERROR | Function | check for an error in a printing operation | 10.3.06  
PRINTER ONLINE | Function | report if printer is on-line | 10.3.06  
PRINTER OPEN | Instruction | open printer device | 10.3.01  
PRINTER OUT | Instruction | print data from an address | 10.3.05  
PRINTER SEND | Instruction | send a string to the printer | 10.3.01  
PRIORITY OFF | Instruction | set Bob priority to default status | 7.2.05  
PRIORITY ON | Instruction | set Bob priority to highest y-coordinate | 7.2.05  
PRIORITY REVERSE OFF | Instruction | toggle off reverse priority of Bobs | 7.2.05  
PRIORITY REVERSE ON | Instruction | toggle on reverse priority of Bobs | 7.2.05  
PROC | Structure | call a procedure | 5.5.02  
PRoc | Embedded Menu Command | call a procedure | 6.5.16  
PROCEDURE | Structure | create a procedure | 5.5.01  
PRUN | Instruction | run a program from memory | 11.4.02  
PU Interface | Instruction | push image to an offset position in the Resource Bank | 9.4.03  
PUT | Instruction | output a record to a random access file | 10.2.15  
PUT BLOCK | Instruction | copy a block onto screen | 7.7.03  
PUT BOB | Instruction | put a fixed copy of Bob onto screen | 7.2.08  
PUT CBLOCK | Instruction | display a compacted block on screen | 7.7.05  
PUT KEY | Instruction | load a string into the keyboard buffer | 10.1.05  
RADIAN | Instruction | use radians | 5.3.08  
RAIN | Reserved Variable | change the colour of a rainbow line | 6.3.05  
RAINBOW | Instruction | display a rainbow | 6.3.05  
RAINBOW DEL | Instruction | delete a rainbow | 6.3.05  
RANDOMIZE | Instruction | set random number seed | 5.3.10  
RDIALOG | Function | read the status of a zone or a button | 9.3.04  
RDIALOG\$ | Function | return text string entered into an edit zone | 9.3.04  
READ | Structure | read data into a variable | 5.4.13  
READ TEXT\$ | Instruction | display a text window on screen | 5.7.06  
REM | Structure | insert a reminder message or comment into program listing | 5.1.1  
REMEMBER X | Instruction | restore the x-coordinate of the text cursor | 5.6.10  
REMEMBER Y | Instruction | restore the y-coordinate of the text cursor | 5.6.10  
RENAME | Instruction | rename a file | 10.2.08  
REPEAT | Structure | mark the start of a conditional loop | 5.4.08  
REPEAT\$ | Function | repeat a string | 5.2.04  
REQUEST OFF | Instruction | cancel the requester | 11.5.06  
REQUEST ON | Instruction | use the AMOS Professional system requester | 11.5.06  
REQUEST WB | Instruction | use the Workbench system requester | 11.5.06  
REserve | Embedded Menu Command | reserve data area for a procedure | 6.5.17

RESERVE AS CHIP DATA | Instruction | reserve a new chip data bank | 5.9.03  
RESERVE AS CHIP WORK | Instruction | reserve a new chip work bank | 5.9.03  
RESERVE AS DATA | Instruction | reserve a new data bank | 5.9.02  
RESERVE AS WORK | Instruction | reserve a new work bank | 5.9.02  
RESERVE ZONE | Instruction | allocate memory for screen zone | 7.4.06  
RESET ZONE | Instruction | erase screen zone | 7.4.07  
RESOURCE\$ | Function | read a message from the Resource Bank | 9.4.03  
RESOURCE BANK | Instruction | select a bank to be used for resources | 9.4.03  
RESOURCE SCREEN OPEN | Instruction | open a screen using resource settings | 9.4.04  
RESOURCE UNPACK | Instruction | unpack an image from the Resource Bank | 9.4.04  
RESTORE | Structure | set the current Read pointer | 5.4.13  
RESUME | Structure | resume program after error trapping | 12.2.02  
RESUME LABEL | Structure | jump to label after error trapping | 12.2.03  
RETURN | Instruction | return from a sub-routine | 5.4.02  
REV | Function | double-flip an image vertically and horizontally | 7.2.12  
RIGHT\$ | Function | give the rightmost characters of a string | 5.2.01  
RND | Function | generate random number | 5.3.10  
ROL.B | Instruction | rotate left the first 8 bits of a value | 14.A.09  
ROL.L | Instruction | rotate left the entire number | 14.A.09  
ROL.W | Instruction | rotate left the bottom 16 bits of a value | 14.A.09  
ROR.B | Instruction | rotate right the first 8 bits of a value | 14.A.10  
ROR.L | Instruction | rotate right the entire number | 14.A.10  
ROR.W | Instruction | rotate right the bottom 16 bits of a value | 14.A.10  
RT Interface | Instruction | return from an Interface sub-routine | 9.2.07  
RU Interface | Instruction | run until conditions are satisfied | 9.1.08  
RUN | Instruction | execute an AMOS Professional program | 10.2.08  
SA Interface | Instruction | save background under a dialogue box | 9.1.08  
SAM BANK | Instruction | change the current sample bank | 8.2.03  
SAM LOOP OFF | Instruction | toggle off repetition loop of sample | 8.2.04  
SAM LOOP ON | Instruction | toggle on repetition loop of a sample | 8.2.04  
SAM PLAY | Instruction | play a sample from the sample bank | 8.2.01  
SAM RAW | Instruction | play a raw sample from memory | 8.2.03  
SAM STOP | Instruction | stop one or more samples playing | 8.2.02  
SAM SWAP | Instruction | activate sample-switching system | 8.2.06  
SAM SWAPPED | Function | test for a successful sample swap | 8.2.06  
SAMPLE | Instruction | assign a sample to the current wave | 8.1.07  
SAVE IFF | Instruction | save an IFF screen to disc | 6.1.11  
SAVE | Instruction | save one or more memory banks to disc | 5.9.03  
SAY | Instruction | speak a phrase | 11.2.01  
SC AMAL | Function | check for Sprite collision | 7.6.09  
SCAN\$ | Function | return a scan-code for use with Key\$ function | 10.1.06  
SCANCODE | Function | give the scancode of a key | 10.1.01  
SCANSHIFT | Function | give shift status of key | 10.1.02  
SCIN | Function | give screen number at hardware coordinates | 6.1.11  
SCREEN | Instruction | set the current screen | 6.1.09  
SCREEN | Function | give the current screen number | 6.1.10  
SCREEN BASE | Function | get screen table | 5.9.11  
SCREEN CLONE | Instruction | clone a screen | 6.1.06  
SCREEN CLOSE | Instruction | eraSWED  
SCREEN COPY | Instruction | copy an area of screen | 6.2.01  
SCREEN DISPLAY | Instruction | position a screen | 6.1.04  
SCREEN HEIGHT | Function | give the current screen height | 6.1.10

SCREEN HIDE | Instruction | hide a screen | 6.1.08  
SCREEN MODE | Function | return screen mode | 6.1.13  
SCREEN OFFSET | Instruction | offset the screen at hardware coordinates | 6.1.05  
SCREEN OPEN | Instruction | open a new screen | 6.1.01  
SCREEN SHOW | Instruction | show a screen | 6.1.08  
SCREEN SWAP | Instruction | swap over the logical and physical screens | 6.2.03  
SCREEN TO BACK | Instruction | move screen to the back of the display | 6.1.09  
SCREEN TO FRONT | Instruction | move screen to the front of the display | 6.1.08  
SCREEN WIDTH | Function | give the current screen width | 6.1.10  
SCROLL | Instruction | scroll a screen zone | 6.2.02  
SCROLL OFF | Instruction | toggle window scroll off | 5.7.04  
SCROLL ON | Instruction | toggle window scroll on | 5.7.04  
SERIAL ABORT | Instruction | stop current data transfer | 10.4.06  
SERIAL BASE | Function | get the address of the serial base | 10.4.06  
SERIAL BITS | Instruction | set the number of bits for transmission of characters | 10.4.02  
SERIAL BUF | Instruction | set the size of the serial buffer | 10.4.04  
SERIAL CHECK | Function | report current serial device activity | 10.4.05  
SERIAL CLOSE | Instruction | close one or more serial channels | 10.4.02  
SERIAL ERROR | Function | report success or failure of last data transfer | 10.4.05  
SERIAL FAST | Instruction | engage fast mode for data transfer | 10.4.04  
SERIAL GET | Function | get a byte from a serial channel | 10.4.03  
SERIAL INPUT\$ | Function | get a string from the serial port | 10.4.04  
SERIAL OPEN | Instruction | open a channel for serial input/output | 10.4.01  
SERIAL OUT | Instruction | output a block of raw data via a serial channel | 10.4.03  
SERIAL PARITY | Instruction | set parity checking for a serial channel | 10.4.02  
SERIAL SEND | Instruction | output a string via a serial channel | 10.4.03  
SERIAL SLOW | Instruction | re-set slow mode for data transfer | 10.4.04  
SERIAL SPEED | Instruction | set the transfer rate for a serial channel | 10.4.02  
SERIAL STATUS | Function | report the status of the Serial Port | 10.4.05  
SERIAL X | Instruction | set handshaking system for serial channel | 10.4.03  
SET ACCESSORY | Instruction | define an accessory program | 13.1.01  
SET BOB | Instruction | set drawing mode for Bobs | 7.3.07  
SET BUFFER | Instruction | set the size of the variable area | 5.1.04  
SET CURS | Instruction | set the shape of the text cursor | 5.6.10  
SET DIR | Instruction | set the directory style | 10.2.04  
SET DOUBLE PRECISION | Instruction | engage double precision accuracy | 5.3.06  
SET ENVEL | Instruction | create a volume envelope | 8.1.08  
SET EQUATE BANK | Instruction | set up the automatic equate system | 11.5.03  
SET FONT | Instruction | select font for use by the Text command | 11.1.02  
SET HARDCOL | Instruction | set hardware register for Sprite collision detection | 7.4.05  
SET INPUT | Instruction | set end-of-line characters | 10.2.13  
SET LINE | Instruction | set a line style | 6.4.03  
SET MENU | Instruction | move a menu item | 6.5.11  
SET PAINT | Instruction | toggle outline mode | 6.4.10  
SET PATTERN | Instruction | select a fill pattern | 6.4.08  
SET RAINBOW | Instruction | define a rainbow | 6.3.04  
SET SLIDER | Instruction | set a fill pattern for a slider bar | 5.7.05  
SET SPRITE BUFFER | Instruction | the the maximum height of Sprites | 7.1.08  
SET STACK | Instruction | set stack space | 5.5.01  
SET TAB | Instruction | change Tab setting | 5.6.08  
SET TALK | Instruction | set the style of synthetic speech | 11.2.01  
SET TEMPRAS | Instruction | set temporary raster | 6.4.11

SET TEXT | Instruction | set the style of text font | 5.6.04  
SET WAVE | Instruction | define a wave form | 8.1.05  
SET ZONE | Instruction | set a screen zone for testing | 7.4.06  
SFont | Embedded Menu Command | set font | 6.5.14  
SF Interface | Instruction | select font to be assigned to text | 9.2.05  
SGN | Function | give the sign of a number | 5.3.04  
SH Interface | Function | read the height of the current screen | 9.2.01  
SHADE OFF | Instruction | toggle text shading off | 5.6.03  
SHADE ON | Instruction | toggle text shading on | 5.6.03  
SHARED | Structure | define a list of shared variables | 5.5.05  
SHIFT DOWN | Instruction | rotate colour values downwards | 6.3.03  
SHIFT OFF | Instruction | turn off colourshifts for current screens | 6.3.04  
SHIFT UP | Instruction | rotate colour values upwards | 6.3.03  
SHOOT | Instruction | generate percussion sound effect | 8.1.01  
SHOW | Instruction | reveal the mouse pointer back on screen | 5.8.03  
SHOW ON | Instruction | reveal the mouse pointer immediately | 5.8.03  
SI Interface | Instruction | define the size of graphics to be saved | 9.1.07  
SIN | Function | give the sine of an angle | 5.3.08  
SL Interface | Instruction | set the style of a line | 9.2.04  
SLine | Embedded Menu Command | set line pattern | 6.5.15  
SLOAD | Instruction | load a section of a sample | 8.2.05  
SM Interface | Instruction | move a screen linked to the mouse pointer | 9.3.17  
SORT | Instruction | sort all elements in an array | 5.2.05  
SP Interface | Instruction | set the fill pattern for a dialogue box | 9.2.03  
SPACE\$ | Function | space out a string | 5.2.04  
SPACK | Picture Compactor | Extension pack a screen | 6.2.05  
SPRITE | Instruction | display a Sprite on screen | 7.1.04  
SPRITE BASE | Function | get Sprite table | 5.9.11  
SPRITEBOB COL | Function | test for a collision between Sprite and Bobs | 7.4.04  
SPRITE COL | Function | test for a collision between Sprites | 7.4.03  
SPRITE OFF | Instruction | remove Sprites from the screen | 7.1.08  
SPRITE UPDATE | Instruction | control Sprite movements | 7.1.08  
SPRITE UPDATE OFF | Instruction | turn off automatic Sprite updating | 7.1.08  
SPRITE UPDATE ON | Instruction | turn on automatic Sprite updating | 7.1.08  
SSAVE | Instruction | save a data chunk anywhere into an existing file | 8.2.05  
SQR | Function | calculate square root of a number | 5.3.06  
SStyle | Embedded Menu Command | set font style | 6.5.15  
START | Function | give the address of a memory bank | 5.9.09  
STEP | Structure | control the increment index in a loop | 5.4.09  
STOP | Instruction | interrupt the current program | 5.1.08  
STR\$ | Function | convert a number into a string | 5.2.03  
STRUC | Reserved Variable | access internal data structure | 11.5.04  
STRUC\$ | Function | read or write a string pointer to a structure | 11.5.05  
STRING\$ | Function | create a new string from an existing string | 5.2.04  
SV Interface | Instruction | set an Interface variable | 9.1.03  
SW Interface | Function | read the width of the current screen | 9.2.01  
SW Interface | Instruction | set the writing mode for text and graphics | 9.2.05  
SWAP | Structure | swap over the contents of two variables | 5.4.06  
SX Interface | Function | get the width of a dialogue box | 9.2.01  
SY Interface | Function | get the height of a dialogue box | 9.2.01  
SYNCHRO | Instruction | execute an AMAL program directly | 7.6.13  
SYNCHRO OFF | Instruction | turn off interrupts | 7.6.13

SYNCHRO ON | Instruction | turn on interrupts | 7.6.13

SYSTEM | Instruction | leave AMOS Professional and go to the Workbench | 5.1.09

SZ Interface | Instruction | save a parameter for the next zone definition | 9.2.09

TAB\$ | Function | move the text cursor to the next Tab position | 5.6.07

TALK MISC | Instruction | set narrator voice | 11.2.02

TALK STOP | Instruction | stop synthetic speech | 11.2.02

TAN | Function | give the tangent of an angle | 5.3.09

TEMPO | Instruction | change the speed of a piece of music | 8.3.02

TEXT | Instruction | print graphic text | 11.1.03

TEXT BASE | Function | give the text base of the current character set | 11.1.03

TEXT LENGTH | Function | give the length of a section of graphical text | 11.1.03

TEXT STYLES | Function | give the current text styles | 5.6.04

TH Interface | Function | return the height of the current font, in pixels | 9.2.05

THEN | Structure | determine action after If | 5.4.03

TIMER | Reserved Variable | count in intervals of 50ths of a second | 5.3.11

TITLE BOTTOM | Instruction | set a title at the bottom of the current window | 5.7.02

TITLE TOP | Instruction | set a title at the top of the current window | 5.7.02

TL Interface | Function | return the number of characters in a string of text | 9.2.05

TO | Structure | mark the end of a loop | 5.4.09

To AMAL | Structure | mark end of a loop | 7.6.05

TRACK LOAD | Instruction | load a Tracker music module | 8.3.02

TRACK LOOP OFF | Instruction | turn off a Tracker module loop | 8.3.03

TRACK LOOP ON | Instruction | loop a Tracker module | 8.3.03

TRACK PLAY | Instruction | play a Tracker module | 8.3.02

TRACK STOP | Instruction | stop all Tracker music | 8.3.03

TRAP | Instruction | trap an error | 12.2.04

TRUE | Function | holds the value of -1 if a condition is true | 5.4.06

TW Interface | Function | return the width of current font text, in pixels | 9.2.05

UI Interface | Instruction | create a user-defined Interface command | 9.2.08

UN Interface | Instruction | unpack an image from the Resource Bank | 9.4.01

UNDER OFF | Instruction | toggle text underlining off | 5.6.03

UNDER ON | Instruction | toggle text underlining on | 5.6.03

UNFREEZE | Instruction | unfreeze the display | 7.5.07

UNPACK | Picture Compactor | Extension unpack a compacted screen | 6.2.06

UNTIL | Structure | mark the end of a conditional loop | 5.4.08

UPDATE | Instruction | move all Objects at once | 7.3.04

UPDATE EVERY | Instruction | control update in | 7.6.12

UPDATE OFF | Instruction | turn off the automatic Object re-drawing system | 7.3.04

UPDATE ON | Instruction | turn on the automatic Object re-drawing system | 7.3.04

UPPER\$ | Function | convert a string of text to upper case | 5.2.03

USING | Instruction | format printed output | 5.6.13

VA Interface | Function | return value held by Interface item | 9.1.03

VAL | Function | convert a string of digits into a number | 5.2.03

VARPTR | Function | read the address of a variable | 14.A.06

VDIALOG | Function | assign or read an Interface value | 9.3.05

VDIALOG\$ | Function | assign or read an Interface string | 9.3.05

VIEW | Instruction | display the current view setting | 6.1.04

VLine Interface | Instruction | draw a vertical line from packed image components | 9.4.02

VOICE | Instruction | activate a voice | 8.3.03

VOLUME | Instruction | control the volume of sound | 8.3.03

VREV BLOCK | Instruction | flip a block vertically | 7.7.04

VREV | Function | flip an image vertically | 7.2.11

VS Interface | Instruction | create an animated vertical slider bar | 9.3.09  
VSCROLL | Instruction | scroll text vertically | 5.6.12  
VSLIDER | Instruction | draw a vertical slider bar | 5.7.05  
VT Interface | Instruction | display vertical text | 9.2.06  
VU AMAL | Function | give intensity of current voice | 7.6.09  
VUMETER | Function | test the volume of a voice | 8.1.09  
WAIT | Instruction | wait before performing the next Instruction | 7.6.16  
WAIT KEY | Instruction | wait for a key-press | 10.1.04  
WAIT VBL | Instruction | wait for the next vertical blank period | 6.2.05  
Wait AMAL | Instruction | turn off main program and wait for Autotest | 7.6.07  
WAVE | Instruction | assign a wave to a sound channel | 8.1.06  
WEND | Structure | mark the end of a conditional loop | 5.4.08  
WHILE | Structure | mark the start of a conditional loop | 5.4.08  
WIND CLOSE | Instruction | close the current window | 5.7.03  
WIND MOVE | Instruction | move the current window | 5.7.03  
WIND OPEN | Instruction | create a window | 5.7.01  
WIND SAVE | Instruction | save the contents of the current window | 5.7.03  
WIND SIZE | Instruction | change the size of the current window | 5.7.04  
WINDON | Function | give the value of the current window | 5.7.03  
WINDOW | Instruction | change the current window | 5.7.01  
WRITING | Instruction | select text writing mode | 5.6.04  
X BOB | Function | give the x-coordinate of a Bob | 7.2.03  
X CURS | Function | give the x-coordinate of the text cursor | 5.6.09  
X GRAPHIC | Function | convert text x-coordinate to graphic x-coordinate | 11.1.04  
X HARD | Function | convert screen x-coordinate to hardware x-coordinate | 7.1.10  
X MENU | Function | give graphical x-coordinate of a menu item | 6.5.10  
X MOUSE | Reserved Variable | give/set x-coordinate of mouse pointer | 5.8.04  
X SCREEN | Function | convert hardware x-coordinate to screen x-coordinate | 7.1.09  
X SPRITE | Function | give x-coordinate of a Sprite | 7.1.09  
X TEXT | Function | convert graphic x-coordinate to text x-coordinate | 11.1.04  
XA Interface | Function | get the previous x-coordinate of the graphics cursor | 9.2.02  
XB Interface | Function | get the current x-coordinate of the graphics cursor | 9.2.02  
XGR | Function | give x-coordinate of the graphics cursor | 6.4.02  
XH AMAL | Function | convert screen x-coord to hardware x-coord | 7.6.10  
XM AMAL | Function | give hardware x-coord of mouse cursor | 7.6.10  
XS AMAL | Function | convert hardware x-coord to screen x-coord | 7.6.10  
XY Interface | Instruction | set graphics variables | 9.2.08  
Y BOB | Function | give the y-coordinate | 7.2.03  
Y CURS | Function | give the y-coordinate of the text cursor | 5.6.09  
Y GRAPHIC | Function | convert text y-coordinate to graphic y-coordinate | 11.1.04  
Y HARD | Function | convert screen y-coordinate to hardware y-coordinate | 7.1.10  
| Y MENU | Function | give graphical y-coordinate of a menu item | 6.5.10  
| Y MOUSE | Reserved Variable | give/set y-coordinate of mouse pointer | 5.8.04  
| Y SCREEN | Function | convert hardware y-coordinate to screen y-coordinate | 7.1.09  
Y SPRITE | Function | give the y-coordinate of a Sprite | 7.1.09  
Y TEXT | Function | convert graphic y-coordinate to text y-coordinate | 11.1.04  
YA Interface | Function | get the previous y-coordinate of the graphics cursor | 9.2.02  
YB Interface | Function | get the current y-coordinate of the graphics cursor | 9.2.02  
YGR | Function | give the y-coordinate of the graphics cursor | 6.4.02  
YH AMAL | Function | convert screen y-coord to hardware y-coord | 7.6.10  
YM AMAL | Function | give hardware y-coord of mouse cursor | 7.6.10  
YS AMAL | Function | convert hardware y-coord to screen y-coord | 7.6.10



Z AMAL |Function| give random number| 7.6.10  
ZC Interface | Instruction | change the status of a zone| 9.3.12  
ZN Interface |Function| return the number of a zone |9.3.14  
ZONE |Function| give zone number under specified screen coordinates |7.4.06  
ZONE\$ |Function| create a zone around text| 5.6.11  
ZOOM | Instruction | change the size of a part of the screen |6.2.03  
ZP Interface |Function| return the status of a zone |9.3.12  
ZV Interface |Function| read a zone variable from the internal buffer area |9.2.09