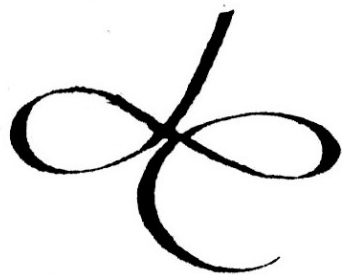


The Professional

Adventure
Writing
System



Quick Reference



Function of P.A.W. Flags

The normal flags are free for use in any way in games. The auto decrement flags (2 to 10) are also free for use, but be sure you know in which situations they are reduced before using them. Other flags should mostly only be set using the appropriate action, but useful tests can be carried out on their contents.

Flag 0 When non zero indicates game is dark (see also object 0)
Flag 1 Holds quantity of objects player is carrying (but not wearing)

The following flags are decremented if non zero by PAW;

Flag 2 When a location is described
Flag 3 When a location is described and it's dark (Flag 0 not 0)
Flag 4 When a location is described, it's dark and object 0 is absent
Flags 5 to 8 Every time frame (i.e. every phrase/timeout)
Flag 9 Every time frame that it's dark
Flag 10 Every time frame that it's dark and object 0 is absent

Flags 11 to 28 are free for use in your own games

Flag 29 holds Picture Control flags
Bit 7 - Set this to force picture to be drawn (LOOK)
Bit 6 - Set this to always draw picture (PICS ON)
Bit 5 - Set this to never draw picture (PICS OFF)
this is set by using the GRAPHIC action.

Flag 30 Score flag
Flag 31/32 (LSB/MSB) holds number of turns player has taken
(actually this is the number of phrases extracted from the players input).

Flag 33 holds the Verb for the current logical sentence
Flag 34 holds the first Noun in the current logical sentence
Flag 35 holds the Adjective for first Noun
Flag 36 holds the Adverb for the current logical sentence

Flag 37 holds maximum number of objects conveyable (initially 4)
Set using ABILITY action.

Flag 38 holds current location of player

Flag 39 holds current top line of screen
Set by the LINE action.

Flag 40 holds screen mode (range 0 to 4..) set with MODE action.
also Bit 7 - Forces no change of Border
Bit 6 - Produces "More.." when screen fills

Flag 41 holds line number for split (if not in range 4-24 then 12 used) this is set by the PROTECT action to be the current screen line.

Flag 42 holds prompt to use (a system message number - 0 selects

one of four randomly
Set by the PROMPT action.

Flag 43 holds the Preposition in the current logical sentence

Flag 44 holds the second Noun in the current logical sentence

Flag 45 holds the Adjective for the second Noun

Flag 46 holds the current pronoun ("IT" usually) Noun

Flag 47 holds the current pronoun ("IT" usually) Adjective

Flag 48 holds Timeout duration required

Flag 49 holds Timeout Control flags

Bit 7 - Set if timeout occurred last frame

Bit 6 - Set if data available for recall (not of use to writer)

Bit 5 - Set this to cause auto recall of input buffer on timeout

Bit 4 - Set this to print buffer on exit, (for use with Bit 3).

Bit 3 - Set this to take input from lower screen

Bit 2 - Set this so timeout can occur on ANYKEY

Bit 1 - Set this so timeout can occur on "More..."

Bit 0 - Set this so timeout can occur at start of input only

Set using INPUT and TIME (as is flag 48), TIMEOUT tests Bit 7 of this flag.

Flag 50 holds Objno. for DOALL loop. i.e. value following DOALL

Flag 51 holds last object referenced by GET/DROP/WEAR/WHATO etc.
This is the number of the currently referenced object as printed in place of any underlines in text.

Flag 52 holds players strength (maximum weight of objects carried and worn - initially 10)
Set with ABILITY action.

Flag 53 holds object print flags

Bit 7 - Set if any object printed as part of LISTOBJ or LISTAT

Bit 6 - Set this to cause continuous object listing i.e. LET 53 64 will make PAW list objects on the same line forming a valid sentence.

Flag 54 holds the present location of the currently referenced object

Flag 55 holds the weight of the currently referenced object

Flag 57 is 128 if the currently referenced object is a container.

Flag 57 is 128 if the currently referenced object is wearable

Flag 58 when set to 128 causes PAW to match words in a sub-process (PARSE usage mainly). Cleared by Process 1/2.

Flag 59 should be avoided as it will be used for any expansion

Flag 60 to 255 are available for your own use.

BORDER	0-7		
CHARSET	0-255		;select character set (if inserted)
SAVEAT			;save current print position
BACKAT			;return to it
PRINTAT	0-20	0-31	;set a new print position
LISTOBJ			;list objects at current location
LISTAT	locno+		;list objects at specified location
INVEN	}		
DESC			;restart main loop, describe HERE
END			;Type 2,exits table to restart game
DONE			;return to caller, done something
NOTDONE			
OK			;done with SM15
SAVE			
LOAD			
RAMSAVE			
RAMLOAD	flagno		
ANYKEY			
PAUSE	0-255		;delay program for n/50 of a second
PARSE			;convert input string to valid LS
NEWTEXT			;force the loss of remaining phrases
BEEP	0-255	0-255	;duration(1/100sec),pitch(BASIC+60/2)
PROCESS	procno		;execute sub-response/process
DOALL	locno+		;generate Noun(Adjective)1 for each object at Location locno.
RESET	locno		;move player and present objects, reset others to initially at - used to chain games with LOAD
EXTERN	0-255		;call external program

Where:

locno. is a valid location number.

locno+ also allows the use of; 252 (not-created), 253 (worn), 254 (carried) and 255 which is converted into the current location of the player.

mesno. is a valid message.

sysno. is a valid system message.

flagno. is any flag (0 to 255).

procno. is a valid sub-process number.

word; is a word of the required type, which is present in the vocabulary, or "-" which ensures no-word - not an anymatch as normal).

The CondsActs

Conditions:

AT	locno		;ensure player at specific location
NOTAT	locno		;or not
ATGT	locno		;higher location than specified
ATLT	locno		;lower...
PRESENT	objno		;specified object is HERE
ABSENT	objno		;or not
WORN	objno		;the object is WORN
NOTWORN	objno		;or not
CARRIED	objno		;etc
NOTCARR	objno		
ISAT	objno	locno+	;tests for an object at a location
ISNOTAT	objno	locno+	; (or not) other than HERE
ZERO	flagno		;the flag contains 0!
NOTZERO	flagno		;guess what.... thats right NOT!
EQ	flagno	0-255	;the flag has a value EQUAL to
NOTEQ	flagno	0-255	;NOTEQUAL
GT	flagno	0-255	;GreaterThan
LT	flagno	0-255	;LessThan
SAME	flagno	flagno	;The SAME as
NOTSAME	flagno	flagno	;or NOT as the other number/flag
ADJECT1	word		;current LS adjective 1 is
ADVERB	word		;etc
PREP	word		
NOUN2	word		
ADJECT2	word		
CHANCE	0-99		;random possibility of success
TIMEOUT			;players last input timed out
QUIT			;Are you sure?

Actions (Those marked { are type 4, } are type 3, | are type 1)

GET	{	objno		;GET specified object
DROP	{	objno		;what they say...
WEAR	{	objno		
REMOVE	{	objno		
CREATE	{	objno		
DESTROY	{	objno		
SWAP	{	objno	objno	;exchanges position.
PLACE	{	objno	locno+	;puts it at the location
PUTO	{	locno+		;puts the current (WHATO) object
PUTIN	{	objno	locno	;for containers
TAKEOUT	{	objno	locno	
DROPALL	{			;all fall down...
AUTOG	{			;auto versions of above

AUTOD			;which do a WHATO etc
AUTOW			
AUTOR			
AUTOP	locno		
AUTOT	locno		
COPYOO	objno	objno	;make 2nd object be with 1st
COPYOF	objno	flagno	;copy position of object to flag
COPYFO	flagno	objno	
WHATO			;convert Noun1(Adjective1) to current object
WEIGH	objno	flagno	;weight of object is put in flag
SET	flagno		;flag becomes 255
CLEAR	flagno		;or 0
PLUS	flagno	0-255	;add value to flag
MINUS	flagno	0-255	;or take away
LET	flagno	0-255	;set to given value
ADD	flagno1	flagno2	;contents of flag1 added to flag2
SUB	flagno1	flagno2	;or subtracted from it
COPYFF	flagno1	flagno2	;duplicated
RANDOM	flagno		;set to random number from 0 to 99
MOVE	flagno		;Adjust contents of flag according to the LS Verb and the Connection table entry for location, that the contents specify. (allows movement in PSIs)
GOTO	locno		;put player at location
WEIGHT	flagno		;weight of objects carried & worn are put in flag
ABILITY	0-255	0-255	;set conveyable objects and strength
MODE	0-4	0-3	
LINE	0-20		;set screen line to split picture
GRAPHIC	0-3		
PROMPT	sysno		;prompt on input. 0 is random
INPUT	0-7		
TIME	0-255	0-7	
PROTECT			;protect text on screen to current print line from scrolling.
PRINT	flagno		;display contents of flag on screen
URNS			;displays no of turns taken
SCORE			;the score
CLS			;clear the screen
NEWLINE			
MES	mesno		;message without a newline
MESSAGE	mesno		;message with a newline
SYSMES	sysno		;system message without newline
PICTURE	locno		;Display picture (without CLS)
PAPER	0-9		
INK	0-9		

The system messages

The majority of system messages contain ESCC 7s to allow correct spacing on the screen.

SM0 - is used instead of the location description when it is dark.
SM1 - is printed by LISTOBJ if at least one object is present.
SM2 to SM5 - are the four input prompts which are selected randomly unless flag 42 is set to be a valid message number.
SM6 - is produced by the parser when no further phrase can be understood.
SM7 - is produced if no action was carried out (or NOTDONE was) in Response when the Verb is < 14
SM8 - is produced if no action was carried out (or NOTDONE was) in Response when the Verb is > 13
SM9 to SM11 - are printed by action INVEN.
SM12 - printed by QUIT
SM13 and 14 - are printed by the END action.
SM15 - the OK action message.
SM16 - the ANYKEY action message.
SM17 to SM20 - are the TURNS action messages.
SM21 and SM22 - are the SCORE action messages.
SM23 to SM29 - are the first of many messages produced by the object manipulating actions.
SM30 - the positive response expected by END and QUIT.
SM31 - the negative response expected by END and QUIT.
SM32 - produced when a screen full of text has appeared.
SM33 - the input marker.
SM34 - the cursor
SM35 - displayed when a timeout occurs
SM36 to SM45 - are more messages produced by the object manipulating actions.
SM46 - the link between objects when listing continuously
SM47 - the final link between the last two objects when listing
SM48 - the termination of a list of objects (printed by both LISTOBJ and LISTAT, so take care.)
SM49 and SM50 - yet more object messages
SM51 - the termination for a compound sentence on PUTIN/TAKEOUT (and AUTOP/AUTOT)
SM52 - a final object message.
SM53 - message for LISTAT action if no objects found.
SM54 - prompt for filename for game position save/load

SM55 onwards are free to be inserted for your own use. PAW on other machines may use more messages, so bear this in mind if you intend transferring the adventure to another version. E.g. PAW under CPM uses messages 55 to 60!

Graphics Editor Commands

Occasionally when moving the line at speed it will disappear, this is due to a beat frequency of update with screen flyback.

Moving the end of the rubber banded line is achieved with:

Q W E	at 1 pixel per move. Alternatively
A D	use Interface 2/Plus 2 joystick port 2
Z X C	or Kempston~ interface (SYMBOL SHIFT & J)

Hold down the CAPS SHIFT key to accelerate to eight pixels per move.

The drawstring editing commands:

Cursor Right	(CAPS SHIFT & 8)	- Start of drawstring
Cursor Down	(CAPS SHIFT & 6)	- Next command
Cursor Up	(CAPS SHIFT & 7)	- Previous command

DELETE	(CAPS SHIFT & 0)	- Delete previous command
GRAPHICS	(CAPS SHIFT & 9)	- Delete next command

All the following commands require SYMBOL SHIFT to be held down unless otherwise specified.

The toggles:

I	Inverse Toggle
O	Over Toggle
Y	Grid Toggle
J	Kempston~ joystick toggle.

The next commands all insert into the database:

A	Absolute Move (a PLOT with I and O selected)
B	Block in rectangle attribute area. Diagonal defined by line
F	Fill area from end of line. (See footnote)
G	Gosub to location number with scale
X	Ink (produces prompt for Ink selection)
L	Line fix (draws rubber banded line)
C	Paper Select
P	Plot point at end of line
R	Relative Move point to end of line
S	Shade an area of screen from end of line with pattern
V	Flash Select
Z	Bright Select
T	Text character on screen, selects character and set no.

Note: Fill and Shade are not completely re-entrant. Fill is a shade with all pixels in pattern set.

ENTER on its own to finish editing session.