

The
Professional



Adventure
Writing
System



A Supplement
for the

Spectrum — Version A 16



PDF Conversion by Colin Woodcock, January 2005

ZXF Magazine - Spectrum Computing Today

Download your free copy from www.cwoodcock.co.uk/zxf

Additional Notes for PAW versions A16 and later.

The current version of PAW is slightly different from that described in the User Guides supplied. The main differences are described below.

User Overlays

Versions of PAW from version A16 are primarily intended to provide a well documented facility for third party software producers to create products which integrate with the PAW system correctly - User Overlays. The products must be written in assembly language and can be up to about 5K in resident length. A document is available from Gilsoft, to genuine interested writers, which describes the function calls and database structure, on provision of a large (AS) S.A.E.

The system is implemented as an extra menu option ('Z'). This is on the second half of the main menu, but can of course be selected whichever is displayed. You will be prompted for which overlay to load. This can be any of the letters A to Z. PAW then searches the current device for an overlay with that extension.

Current Device

PAW now maintains the idea of a 'current device', that is the device (Disc/Cassette/Microdrive), to which data is to be saved and loaded from. Only on the various Disc versions is there a way to change the current device (option Y, described in the additional notes for each drive) from Disc to Cassette and vice-versa. This allows user overlays to take advantage of disc versions to do any saving and loading without actually knowing about the types of drive available to PAW.

Printer Drivers

Printer drivers are now limited to 48 Bytes at address 29587 (PRTADD). The memory has been considerably reduced in order to provide the extra facilities.

The Parser

One or two subtle changes have taken place in the string handling section of the parser from Version A16 onwards. They are designed to provide a facility for multiple commands to be given to PSI's, a facility suggested by Gerald Kellett.

The three changes are as follows, and although they may not seem very major the logic changes they represent could affect some games if you weren't aware of them:

1/ The PARSE CondAct now maintains a 'current position' within the string in the current logical sentence. Thus a second PARSE CondAct will continue from where the last left off. Previously a subsequent PARSE would have given the same logical sentence as

the first. Thus SAY TO PSI "GET SWORD AND CLEAN IT" can now be made to work with some processing as described below

2/ The PARSE action does not now affect the 'command line empty/valid' flag - the one set by NEWTEXT. This means that a statement such as; SAY TO PSI "HGGHHG". GET SWORD - will now continue on to do the GET SWORD action. Prior to this version the NEWTEXT flag would have been set automatically. This was changed to allow multi-parsing to find the last command in a string without always setting the flag. You will thus need to add a NEWTEXT action to old games just after the PARSE CondAct - which is where processing goes if the string was invalid or empty - if the games are to operate identically with the new paws.

3/ The current verb and Adverb are not cleared (flags 33 and 36 to 255) when a string is parsed (i.e. the PARSE action). This means that if a Verb (or Conversion Noun) is omitted from the first phrase in the string then the current verb will be the one from the phrase which triggered the PARSE (usually SAY or TALK). This is a minor change which means that the current Verb is maintained when the string is multi-parsed. I.e. SAY TO PSI "GET SWORD AND SHIELD" will now work with the processing shown below.

Extra System Flag

Flag 58 has now achieved the status of a system flag... Don't say you've used it - it was marked as 'reserved for expansion'.

If you set this flag to 128, in a Process table, PAW will start to match words which it normally doesn't do except in Response. This allows the multi-parse facility to provide actions for a PSI during Process 2. It will also have other uses we are sure...

The effect is cancelled next time Process 1 or 2 are carried out - by PAW subtracted 128 from the flag. This ensures that Process 1 and 2 act as normal until specifically told to change. You can of course cancel the effect yourself by setting the flag back to zero.

Using Multi-Parse

The ability to give a PSI a list of commands to do has incredible possibilities for the creation of synchronized problems. Where both the PSI and the player must work together.

These sort of problems can add a whole new dimension to adventures and are well worth considering, here are some suggestions:

Imagine a game with a room that is instant death for the player which contains an object that he requires. You could instruct a PSI to go in, get the object and come back out.

Say that in order to kill a certain monster you needed a simultaneous attack from three characters. You could use

the following:

SAY TO PSI1 "WAIT, KILL MONSTER"

SAY TO PSI2 "KILL MONSTER"

KILL MONSTER

All three KILL MONSTERS would be carried out in the same time frame.

They say the best way to demonstrate something is by example. So here goes with a short listing of a game with only one problem:

In order to get out of a cavern you need to be lifted on a platform controlled from another room. This can only be achieved by giving a PSI (who happens to be hanging around) a list of things to do. I.e. Go to the cavern and pull the rope. While you in the meantime step onto the platform and wait...

Flag Usage

20 - Location of PSI

21 - copy of flag 20 during movement processing

60 - when 0 indicates platform is on floor, 1 - held by PSI and 2 - held by Player.

195 - Players Verb/Pronoun-Noun Saved

196 - Players Adverb/Pronoun-Adjective Saved

197 - Number of Logical Sentences waiting for PSI

198 - Next storage flag group to store LS in

199 - Next storage flag group to get a LS from

200 - 206 - Store 0 for LS

207 - 213 - Store 1 for LS

214 - 220 - Store 2 for LS

Notes

The principle of the multi-parse is that the entire string is broken down into a list of LSs that the PSI will be required to do. These LSs are then stored (saved if you like) in some flags to be doled out, one per timeframe (use of process 2).

The LSs waiting for the PSI to do are held in a 'queue' which is a computer term for an ordered list. They are actually held in a 'round robin fifo queue'. fifo stands for 'first in first out'. i.e. the first LS given to the PSI must be the first it carries out. While 'round robin' indicates that the LS storage used goes around the available storage flags in a circular motion. i.e. it goes back to the beginning when it falls off the end! Thus the groups of flags will be used in the order; Store 0, Store 1, Store 2, Store 0 etc. The use of only three storage areas means that only three commands can be queued for the PSI, there is no reason why this cannot be expanded upon. Indeed if you only needed Verb Noun commands to be given to PSI's you could save only those parts of the LS. Thus requiring only two flags per LS not 7!

The extraction of multiple phrases is done by a single process table which calls itself to get the next phrase. This is known as 'recursion' and is simpler than a sequence of entries doing PARSE and PROCESS calls etc. It does limit you to 9 phrases in a string though - Why? (Clue: you can only nest PROCESS calls to a depth of ten.)

Locations

Location 0

I am in a large cavern. On the East wall, high up, is an entrance from which a shaft of light descends. A lifting platform, obviously intended as a means of getting to the entrance, is linked via a series of pulleys on the roof to a steel cable which disappears into a hole in the North wall just above a tunnel.

Location 1

I am standing on a platform

Location 2

I am standing on a ledge overlooking a lush green valley. To the West is an entrance to a large cavern.

Location 3

I am in a small ante-room. A twisting tunnel leads South. A steel cable hangs from the ceiling.

Connections

Location 0 N TO 3

Location 1

Location 2

Location 3 S TO 0

Messages

Message 0

A PSI is here.

Message 1

The PSI doesn't understand.

Message 2

You have said enough to the PSI.

Message 3

You speak to the PSI.

Message 4

The PSI cannot do that.

Message 5

The PSI pulls on the cable.

Message 6

The PSI releases his grip on the cable.

Message 7

The PSI stands on the platform.

Message 8

The PSI steps off the platform.

Message 9

The PSI leaves.

Message 10

A platform

Message 11
 The platform
 Message 12
 rests on the floor of the cavern.
 Message 13
 hangs just inside the opening.
 Message 14
 now
 Message 15
 which
 Message 16
 jars into motion.
 Message 17
 A PSI arrives.
 Message 18
 The PSI can't go that way.
 Message 19
 You release your grip on the cable.

Response Table

*	*	EQ	60	2 ;Player holding cable?
		CLEAR	60	;Release it
		MESSAGE	19	
		PROCESS	8	;Cancel DONE flag
I	_	INVEN		
GET	PLATF	PREP	OFF	;Movements on and off platform
		AT	1	
		ZERO	60	
		GOTO	0	
		DESC		
GET	PLATF	PREP	OFF	
		AT	1	
		GOTO	2	
		DESC		
GET	PLATF	PREP	ON	
		AT	0	
		ZERO	60	
		GOTO	1	
		DESC		
GET	PLATF	PREP	ON	
		AT	2	
		NOTZERO	60	
		GOTO	1	
		DESC		
R	_	DESC		
QUIT	_	QUIT		
		URNS		

```

                                END
SAVE      _      SAVE
LOAD      _      LOAD
RAMSA     _      RAMSAVE
RAMLO     _      RAMLOAD 255
SAY       PSI    NOTSAME 20    38 ;Talk to PSI if in cavern
          PSI    ATLT    2      2 ;or on platform etc
          PSI    LT      20    2
          PSI    PROCESS 3
          PSI    DONE
SAY       PSI    SAME    20    38 ;otherwise have to be same
          PSI    PROCESS 3      ;location
          PSI    DONE
SAY       PSI    MESSAGE 20
          PSI    DONE
WAIT      _      OK
PULL      CABLE  AT      3      ;Allow player to hold cable
          CABLE  ZERO    60
          CABLE  LET     60    2
          CABLE  OK
RELEA     CABLE  OK
STAND     PLATF  PREP    ON
          PLATF  AT      0
          PLATF  ZERO    60
          PLATF  GOTO    1
          PLATF  DESC

```

Process 1

```

*         *         EQ      31    0 ;Deal with start of game
          *         EQ      32    0
          *         MODE    1     1 ;Continuous scrolling text
          *         TIME    8     3 ;Timeouts
          *         INPUT   7     ;Input at bottom of screen
*         *         NEWLINE ;Always start a fresh line
          *         ATLT    2     ;In cavern or on platform
          *         MES     11    ;"The Platform"
*         *         AT      2     ;Outside cavern
          *         NOTZERO 60    ;Platform is at top
          *         MES     10    ;"A Platform"
*         *         ATLT    2     ;In cavern or on platform

```


		ZERO	60		;which is on floor
		MESSAGE	12		;" rests on the floor."
*	*	ATLT	3		;Anywhere except anti-room
		NOTZERO	60		;Platform at entrance
		MESSAGE	13		;" by the entrance."
*	-	ZERO	0		;Standard PAW dark stuff
		ABSENT	0		;for Object list
		LISTOBJ			
*	-	PRESENT	0		
		LISTOBJ			
-	-	SAME	20	38	;PSI where player is?
		MESSAGE	0		;"There is a PSI here."

Process 2

*	*	NOTZERO	197		;Any commands for PSI
		LET	58	128	;Allow word matching
		PROCESS	5		;extract next action for PSI
		CLEAR	58		;Prevent word matching

Process 3 - Deals with speech to PSI

*	*	COPYFF	46	195	;Save 'IT' for player
		COPYFF	47	196	
		SET	46		;No IT at mo!
		SET	47		
		PARSE			;Get a phrase
		MESSAGE	1		;nOt one there
		COPYFF	195	46	;Restore IT
		COPYFF	196	47	
		DONE			;all over
*	*	MESSAGE	3		;"You speak to PSI"
		PROCESS	4		;extract and store phrases
		COPYFF	195	46	;Restore IT
		COPYFF	196	47	

Process 4 - This will extract and store up to three phrases although this could be expanded with a few simple changes/extra entries. Note that this is Recursive as it calls itself!

*	*	EQ	197	3	;Max of three phrases in queue
		MESSAGE	2		;"Said enough to PSI."
		DONE			
*	-	ZERO	198		;Use store 07
		COPYFF	33	200	
		COPYFF	34	201	
		COPYFF	35	202	
		COPYFF	36	203	

		COPYFF	43	204	
		COPYFF	44	205	
		COPYFF	45	206	
*	-	EQ	198	1	;Use store 1?
		COPYFF	33	207	
		COPYFF	34	208	
		COPYFF	35	209	
		COPYFF	36	210	
		COPYFF	43	211	
		COPYFF	44	212	
		COPYFF	45	213	
*	-	EQ	198	2	;Use store 2?
		COPYFF	33	214	
		COPYFF	34	215	
		COPYFF	35	216	
		COPYFF	36	217	
		COPYFF	43	218	
		COPYFF	44	219	
		COPYFF	45	220	
*	-	PLUS	197	1	;One more phrase stored
		PLUS	198	1	;Next store
		EQ	198	3	;reached the last?
		CLEAR	198		;Go back round
-	-	PARSE			;Get another phrase
		DONE			;No more there so finished
-	-	PROCESS	4		;Store it
Process 5	-	Extracts the next phrase from store for the PSI			
*	*	COPYFF	33	195	;Save Verb/Adverb of player
		COPYFF	36	196	
*	-	ZERO	199		;Store 0?
		COPYFF	200	33	
		COPYFF	201	34	
		COPYFF	202	35	
		COPYFF	203	36	
		COPYFF	204	43	
		COPYFF	205	44	
		COPYFF	206	45	
*	-	EQ	199	1	;Store 1?
		COPYFF	207	33	
		COPYFF	208	34	
		COPYFF	209	35	
		COPYFF	210	36	
		COPYFF	211	43	
		COPYFF	212	44	
		COPYFF	213	45	

*	-	EQ	199	2	;Store 2?
		COPYFF	214	33	
		COPYFF	215	34	
		COPYFF	216	35	
		COPYFF	217	36	
		COPYFF	218	43	
		COPYFF	219	44	
		COPYFF	220	45	
-	-	MINUS	197	1	;One less in store
		PLUS	199	1	;Extract next from one more
		EQ	199	3	;Reached end?
		CLEAR	199		;Back to start
-	-	PROCESS	6		;Do the job
		COPYFF	195	33	;Restore player Verb/Adverb
		COPYFF	196	36	

Process 6 - Commands that can be given to PSI

*	*	EQ	60	1	;Holding Cable?
		AT	3		;Where player can see PSI?
		MESSAGE	6		;"PSI Releases grip"
*	*	EQ	60	1	;Holding cable?
		CLEAR	60		;Release grip.
		ATLT	3		;Can player see effect?
		MES	11		;Describe "The platform"
		MESSAGE	16		;" jars into motion."
		MES	11		;"The platform"
		MES	14		" now"
		MESSAGE	12		;" rests on the ground."
GET	PLATF	PREP	OFF		;GET OFF PLATFORM
		EQ	20	1	;PSI on it?
		ZERO	60		;Platform on ground?
		CLEAR	20		;Put PSI in cavern (loc 0)
		ATLT	2		;Can player see it?
		MESSAGE	8		;"PSI steps off."
		DONE			
GET	PLATF	PREP	OFF		;GET OFF PLATFORM
		EQ	20	1	;PSI on it?
		LET	20	2	;Platform by entrance?
		ATLT	2		;Player see it?
		MESSAGE	8		;"PSI steps off."
		DONE			
GET	PLATF	PREP	ON		;GET ON PLATFORM
		ZERO	20		;PSI on ground?
		ZERO	60		;along with platform?
		LET	20	1	;Move PSI to platform
		ATLT	2		;Can player see it?
		MESSAGE	7		;"PSI steps on."
		DONE			

PULL	CABLE	EQ	20	3	; PSI in anti-roan?
		ZERO	60		;with no one holding cable?
		AT	3		;Is player here as well?
		LET	60	1	;PSI holding cable
		MESSAGE	5		;"PSI grips cable."
		DONE			
PULL	CABLE	EQ	20	3	; PSI in anti-room?
		ZERO	60		;with no one holding cable?
		ATLT	2		;Can player see result?
		LET	60	1	;PSI holding cable
		MES	11		;Describe "The platform"
		MESSAGE	16		" jars into motion."
		MES	11		;"The platform"
		MES	14		" now"
		MESSAGE	13		;" hangs by the entrance."
		DONE			
RELEA	CABLE	DONE			; Is done by any action!
STAND	PLATF	PREP	ON		;STAND ON PLATFORM
		AT	0		;See above GET ON PLATFORM
		ZERO	60		
		LET	20	1	
		MESSAGE	7		
		DONE			
WAIT	—	DONE			;Do nothing for a time frame
—	—	LT	33	14	;Movement?
		PROCESS	7		;Deal with it
		DONE			
—	—	CLEAR	197		;Can't do it so cancel any
		CLEAR	198		;waiting LS for PSI.
		CLEAR	199		
		SAME	20	38	;Is player where PSI is?
		MESSAGE	4		;"PSI can't do it."
Process 7 - Deal with movement for PSI					
*	*	COPYFF	20	21	;Save current location
		MOVE	20		;Try and move
		NOTSAME	20	21	;Did location change?
		SAME	21	38	;Was player there?
		MESSAGE	9		;tell them "PSI leaves."
*	*	NOTSAME	20	21	;Somewhere new?
		SAME	20	38	;Where player is?
		MESSAGE	17		;tell them "PSI arrives."
*	*	SAME	20	21	;No change?
		CLEAR	197		;Can't go that way so
		CLEAR	198		;clear any outstanding LS

```
CLEAR      199      ;for PSI
SAME       20      38 ;Player here?
MESSAGE    18      ;tell them.
```

Process 8

```
*      *      NOTDONE      ;Cancel the 'done' flag
```

Playing

If you do type this in you may like to try some of the following sequences from the starting position...

```
GET ON PLATFORM, SAY TO PSI "GO NORTH, PULL CABLE AND RELEASE IT"
THEN GET OFF IT.
```

This shows the independence of IT for Player and PSI.

```
SAY TO PSI "N, PULL CABLE", STAND ON PLATFORM, GET OFF IT
```

Is the solution, although if you wished to lower the platform after.

```
SAY TO PSI "N, PULL CABLE & RELEASE IT",GET ON PLATFORM AND OFF IT
```

Would leave you outside without a platform, while...

```
SAY TO PSI "STAND ON PLATFORM, WAIT THEN GET OFF IT". N, PULL
CABLE, RELEASE IT, S
```

Would leave you without a means of exit and the PSI outside!

A new overlay - Hunk Management

The latest versions of PAW are supplied with an extra overlay, implemented under the user overlay scheme. This is overlay H - Hunk Management. It is supplied for your convenience as a useful utility. It allows the manipulation of the data which may be inserted in the database by other user overlays. This data is inserted in a documented fashion by well behaved User Overlays using a system of memory Hunks (sections or areas of the database). The hunks of memory can be almost any size from 0 bytes (there is always a 3 byte overhead so a zero byte hunk will be three bytes long - it just won't have any room for information!) to the size of the free memory (although on a 128K Spectrum the maximum size of all hunks is limited to about 6K if you wish to use other character sets).

Each user overlay may own one (or more) hunks to contain information which will be preserved with the database. An example of this is the Direction Pointer Table (DPT) of PAW-TEL (one of the PTM overlays) which is used to describe how the various directions will be represented with the Map command. Thus it is related to the database and is included within it to save retyping it every time you load PAW-TEL.

The Sub-Menu

Hunk management is presented in the same format as other PAW menus. In the following description of each command, 'overlay' indicates the letter of the User Overlay which 'owns' the hunk. E.g.. The DPT would be owned by overlay 'T' as it is used by PAWTEL.

Insert **I overlay size**

Will insert a hunk of space (and initialize it to zero) of size (plus three byte overhead) belonging to User Overlay overlay. Thus to insert a DPT (for PAW-TEL) you would use I T 12, to insert the required space - This will of course insert 15 bytes, 12 of which are for data.

Delete **D overlay (n)**

Will allow the n(th) hunk belonging to User Overlay overlay to be deleted. It is possible (but not usual) for a User Overlay to own more than one Hunk, this allows you to delete the required one!

Load **L overlay (n)**

Allows a file to be loaded from the current device into the data area of the n(th) hunk belonging to User Overlay overlay. It must load exactly the right number of bytes (E.g. 12 for a DPT) to fill the data space of the hunk.

Save **S overlay (n)**

Allows the data area of the n(th) hunk belonging to User Overlay overlay to be saved to the current device.

Verify **V overlay (n)**

Will allow the data area of the n(th) hunk belonging to User Overlay overlay to be compared against a file on the current device. This is only of use if no change has occurred in the address of the hunk, i.e. soon after Saving it!

Print P

Will list any hunks present in the database, as the User Overlay with 'owns' them, which number they are and their true size - i.e. including the three byte overhead. Thus the DPT would be represented as:

T	0	15
---	---	----

There is no theoretical limit to the number of hunks belonging to a User Overlay, but a practical limit is set by free memory and the fact that Hunk Management can handle a maximum of 255!

Uses

The Hunk Management overlay will have no direct use immediately, but as more user overlays become available (or you write some yourself) you will find it useful to keep track of data being handled by the overlays. Some suggestions follow:

1/ Some user overlays may provide no way to Save and Load the data from their hunks to use in other databases. PAW-TELS' DPT is a trivial example. You could use the Hunk Management to do this using its Save and Load commands.

2/ Indeed if they are feature packed some overlays may provide no way of Inserting a data area for themselves - again this can be achieved with Hunk Management.

3/ Perhaps the most useful is to allow you to squeeze the last useful features into your game, by deleting all the unnecessary Hunks as you approach a full database!



© 1980 Gilsoft International Ltd.

Published by Gilsoft International Ltd.,
2 Park Crescent, Barry, South Glamorgan CF6 8HD
Telephone Barry (0446) 732765



All rights reserved, unauthorised copying, hiring or lending strictly prohibited