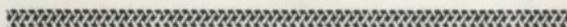
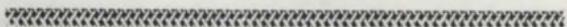


PROAW

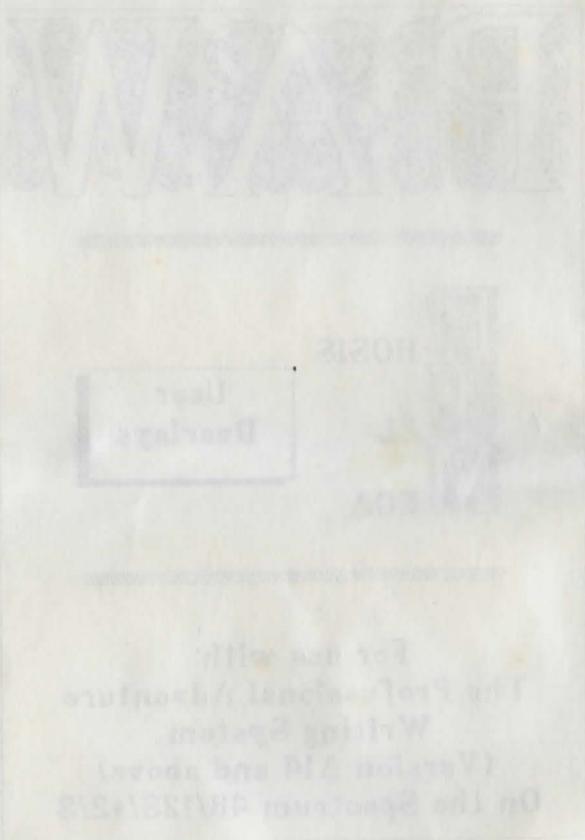


PHOSIS
REL
MEGA

User
Overlays



For use with:
The Professional Adventure
Writing System
(Version A14 and above)
On the Spectrum 48/128/+2/3



PTM - PAW-PHOSIS/TEL/MEGA

Product Design & Concept: (c) 1987 G.T.Kellett T/A KelSoft
Software implementation: (c) 1987 G.T.Kellett T/A KelSoft
Documentation: (c) 1988 Gilsoft International Ltd.
Published by Gilsoft International Ltd. 2 Park Crescent,
Barry, South Glamorgan, CF6 8HD.

Introduction.

PHOSIS, TEL and MEGA are the first three User Overlays available for Professional Adventure Writer. Note that you require a version A14 PAW or later with Option Z - User Overlays - on the main menu, to use them. If you don't have one of these versions, Gilsoft have a standard upgrade available for a charge of £2.99 from any previous version. Please remember to return your Disc if you have a +3 disc version. DISCiPLE/MicroDrive/OPUS etc are supplied on cassette with a transfer utility - so there is no need to return your disc.

Loading.

In order to load one of the overlays use Option Z from PAW's main menu and reply with one of the following to the prompt:

- P - To load PHOSIS
- T - To load TEL
- M - To load MEGA

You will then be presented with a sub-menu which has several numbered options. They are numbered to prevent mistakes, as some sub-menu options are repeats of main menu options. The sub-sub-menus available from this sub-menu follow the PAW format though - phew!

The overlays are recorded in the order above on the cassette, but note that they are preceded by a small utility to transfer them to disc. If you have a disc version you can load this using LOAD "", it will autorun and prompt you for the drive to copy to. It would be a good idea to save them onto the same drive as your working copy of PAW, to reduce disc changes when working. All the Save/Load (and Verify with Cassette/Microdrive) options will now use the currently selected media - as given by option Y on PAW's main menu.

Paw Options

Several of the options on the sub-menus of PHOSIS and TEL are a repeat of main menu options, provided to reduce Overlay swapping. These perform in much the same way as the corresponding options from PAW's main menu. The main difference is that option Z is return to sub-menu not main menu. The options provided are the most common you will carry out while working with PHOSIS and TEL, if you use these in preference to the main menu options then you will not need to reload the PHOSIS or TEL overlay to reselect the Auxiliary sub-sub-menu. The only safe main menu options to use which do not use an overlay are SAVE/LOAD/VERIFY Database - and Free Memory on a cassette version.

Note that PHOSIS allows the Object words to be edited as a sub - menu option, but due to space restrictions this was omitted from

TEL, as was Begin New Table on the Process option, as the code was just too big to fit in.

The Process and Response Tables.

All the overlays consider the Response table to be Process 0 - which indeed it is, internally to PAW. This means that the Process and Response options are also slightly different:-

Response

This merely selects Process 0, which is actually what Response is, and calls the Process option.

Process

Will go back to the process table number that you last worked on - unless you have selected Response in which case you will be returned to Process 0 (Response!) - which will then have been the last Process table you worked on!

Note that 'worked on' for PHOSIS also includes the Process specified for D,P,S,V or L, and the Destination for the C and T commands on the Auxiliary menu.

If you load a database with fewer process tables, making the current table number invalid, the overlays will select table 2.

The Command Line.

The command lines for TEL and PHOSIS are fairly complex. On screen reminders of the possibilities are provided in a compact form. This manual contains a description of the possibilities in a standard format used on larger computer systems to describe command line options. The symbols used are as follows:-

- [] - indicates a compulsory item
- { } - indicates an optional item
- | - indicates an OR decision (i.e. only one of the possibilities can be used)

It is worth understanding these descriptions as they are a compact way of describing the options available. Lets look at an example from PHOSIS in detail. The Copy and Transfer commands both have the following description:

[Process Verb Noun n] [=][[:Process] [=n]][[Verb Noun {n}]

Thus, there are three compulsory items required on the command line - some of which contain separate items of their own. Note that the last two groups could be one of two possibilities each, indicated by the square brackets and the OR symbol ([:][]), one of the two options MUST be present for both items.

The first group requires a Process table number, a Verb and Noun and an entry number. This effectively describes an entry in any Process table precisely - Remember that Response is Process 0.

This group gives the 'source' entry - the one to be copied or transferred (moved).

The second group gives the first part of the 'destination' -where the source will be transferred or copied to. This can be either an equals sign (=) which indicates the same table number, or a colon (:) followed by another process table number. This allows the entry to be moved/copied within the same table or between different tables.

The third group gives the final part of the destination, by specifying the entry within the table. This can be an equals sign (=), specifying the same Verb and Noun as before, optionally followed by a number, indicated by the braces ({}), which specifies a specific entry number - as opposed to the last entry which is where it will go otherwise! Alternatively you can specify a new Verb and Noun entry to move too, which again can be followed by an optional entry number.

The section on the Copy and Transfer commands in the description of PAW-PHOSIS gives some examples of valid command lines.

Streams.

Both TEL and PHOSIS allow a stream to be specified as part of some of their commands. This will cause the output to be sent to the given stream. The most obvious use of this is to use a printer by specifying £3 - this is what the L(Print) options on PAW do. The ability to specify a stream will allow you to use several devices e.g. Serial and ZX Printers at the same time, if you open channels to them before loading PAW.

Examples.

The examples given in the following sections give only a sample of the possibilities that the commands offer. They are often some of the most useful, but, by no means are they the only things you can do - experimentation is the best way of learning anything...

PHOSIS

PAW-PHOSIS is designed to allow the manipulation of Process tables, to provide additional powerful editing facilities for PAW. The available commands can be broken into two groups:

- 1/ The manipulation of individual and groups of entries in tables. This allows entries to be examined, deleted, copied and transferred.
- 2/ The storage and recall of entire Process tables. Allowing the transfer and copying of tables within or between databases.

Auxiliary menu

Note that in all of the following commands, the first entry specified must exist in the table, or the command will merely return 'Not found'.

COPY

Format:

C [Process Verb Noun n] [=][:Process] [=n]][Verb Noun {n}]

Description:

Will duplicate the given entry within the same table or to a different table, either under the same Verb Noun pair or to a different Verb Noun pair. An entry number may be specified for the destination, if none is given the last entry of that Verb Noun pair is assumed.

Examples:

C 0 GET LAMP 0 = GET TORCH

Will make a copy of the CondAct list of entry 0 in the GET LAMP word pair in Response (Process 0 = Response) into the last entry (creating a new entry along the way) of the GET TORCH word pair also in Response.

C 0 GET LAMP 1 :10 =

Will make a copy of entry 1 in the GET LAMP word pair in Response into a new last entry for GET LAMP in Process 10.

C 3 GET LAMP 8 :0 GET TORCH 3

Will make a copy of entry 8 in the GET LAMP word pair in Process 3 into a new entry 3 for GET TORCH in Response.

Notes:

There is no block (or group) Copy option with PHOSIS. But the recall last command (=) key can be used to reduce the typing needed. Merely recall the last line and edit the source entry number (and the destination entry number if one given) to one greater than before and re-enter.

If you have to Copy all or most of a Process table, instead of using Copy you may be quicker saving and reloading the entire table, editing the entries as required afterwards.

If you have a number of similar entries to input, you can save yourself some typing by using Copy to duplicate one entry, then editing the duplicates for any slight differences.

TRANSFER

Format:

T [Process Verb Noun n] [=][:Process] [=n]][Verb Noun {n}]

Description:

Will duplicate the given entry within the same table or to a different table, either under the same Verb Noun pair or to a different Verb Noun pair. An entry number may be specified for the destination, if none is given the last entry of that Verb Noun pair is assumed. The original entry is deleted.

Examples:

T 0 GET LAMP 0 = GET TORCH

Will make a copy of the ConDact, list of entry 0 in the GET LAMP word pair in Response (Process 0 = Response) into the last entry (creating a new entry along the way) of the GET TORCH word pair also in Response. The original GET LAMP entry will be deleted.

T 0 GET LAMP 1 :10 =

Will make a copy of entry 1 in the GET LAMP word pair in Response into a new last entry for GET LAMP in Process 10. The original GET LAMP entry in Response will be deleted.

T 3 DROP LAMP 8 :0 DROP TORCH 3

Will make a copy of entry 8 in the DROP LAMP word pair in Process 3 into a new entry 3 for DROP TORCH in Response. The original entry 8 of DROP LAMP will be deleted from Process 3.

Notes:

The T(transfer) option is useful when changing word numbers in the vocabulary. If for example you needed to change the word value of a Noun so that it was a real noun - having perhaps accidentally inserted it as a proper noun less than 50 - but did not want to lose all the entries you had laboriously inserted; You could insert a synonym for the word such as TEMP etc. Delete the real word and reinsert it at its new word value. You now have the entries preserved under GET TEMP or whatever. Now all you need to do is T(transfer) each entry from GET TEMP to GET word.

There is no block (or group) Transfer option with PHOSIS. But the recall last command (=) key can be used to reduce the typing needed. Merely recall the last line and edit the destination entry number, if one given, to one greater than before and re-enter. There is no need to change the Source entry number as it will be correct each time, as the previous owner of that entry number has been deleted!

If you have to Transfer all or most of a Process table, instead of using Transfer you may be quicker saving and reloading the entire table, deleting the old and editing the new entries as required afterwards.

DELETE

Format:

D [Process Verb Noun n] [=n]][Verb Noun {n}]

Description:

Will delete (remove) all the entries from the given Process table (remember that Response is Process 0), from the specified entry number of the first Verb Noun, to either the last or specified entry number of the second Verb Noun word pair, inclusive.

Examples:

D 0 GET LAMP 0 =

Will delete all the GET LAMP entries in the Response table

D 4 GET LAMP 1 =3

Will delete entries 1,2 & 3 of GET LAMP in Process 4.

D 2 * * 0

Assuming there was at least one * * entry in Process 2, would delete all the entries in the table!

D 0 GET TORCH 0 = DROP TORCH 2

Would delete all the entries between GET TORCH 0 and DROP TORCH 2 in Response.

PRINT

Format:

P [#stream] [Process Verb Noun n] [=n]][Verb Noun {n}]

Description:

Will print (display) all the entries from the given Process table (remember that Response is Process 0), from the specified entry number of the first Verb Noun, to either the last or specified entry number of the second Verb Noun word pair, inclusive, to the screen or given stream number.

Examples:

P 3 GET LAMP 0 =

Will display all the GET LAMP entries in Process 3.

P £3 0 DROP TORCH 2 =3

Will display entries 2 and 3 of DROP TORCH in Response on the device connected to Stream 3 - usually a printer.

P 0 GET TORCH 0 DROP TORCH

Will display all entries of GET TORCH and DROP TORCH, along with any entries in between in Response.

SAVE/VERIFY/LOAD Process table files (PTF's)

These commands can be used to save verify and load a Process table as an individual entity on the current device. They are stored as the current command and so recall ('=') will work to allow fast verify of saved files without retyping the entire command line.

SAVE

Format:
S Process Name

Description:

Saves the given Process table to cassette/disc with filename Name. Note that the name is given as part of the command line. The saved table is called a Process Table File or PTF.

VERIFY

Format:
V Process Name

Description:

Verifies the give Process table against a PTF file on cassette/disc with filename Name. Note that the Process table must be at the same address in memory as the one on tape. I.e. It is only really of use to Verify the table just saved, before making any changes to the database.

LOAD

Format:
L Process Name

Description:

Loads filename Name, to become the specified Process table. If the table is currently occupied by one or more entries you will be prompted to delete the table contents. Any reply other than 'Y' will abort the Load leaving the current contents of the table intact. If you confirm the deletion with 'Y' then the table will be made null, there is no way to recover the contents unless you have a saved database or PTF with the table in - so be careful!

If you are using a 48K Spectrum (Or a 128K in 48K mode) you will be prompted to preserve the overlays if they are currently in memory - this is needed as PHOSIS does not know how large the PTF is going to be and so it opens up the entire database. If there is insufficient memory in the database, or a loading error occurs, the load will abort and the table will be null.

As there is no way of ensuring that all the words used in one database will exist in any other, each Verb Noun pair is tested for validity in the context of its new database. If any word used in not found then all word pairs have to be rendered to those that will always exist, I.e. * and . The first 256 entries of a LOAded table will be made * * entries, the next 256 * _ entries, next 256 _ * and finally the last 256 _ _ ! For this Reason any

table with 'missing words' which have over 1024 entries will be aborted with a 'Limit Reached' error. SAVE does not prevent you from saving tables with more than 1024 entries as it is possible that all the saved words will exist in the new database. It is advisable though to limit saved PTF's to less than 1024 entries to ensure they can be loaded into any database.

As a related problem the parameters of various CondActs are checked to ensure they are valid in the context of the new database:

Any words used in Conditions (e.g. ADJECT1, PREP etc.) that do not have a correspondingly numbered word of the correct type will be converted to *. PAW does not usually allow * in this context, they stand out in listings and will prevent the re-entering of the CondAct list. In addition the Condition will always fail! Note that any words in the new database will correspond only in word number and type to those in the original database, they will not necessarily be the same words.

The parameters to any PROCESS actions will be converted to '2' if a corresponding table does not exist in the database!

Any CondActs with a parameter of Location, Object, Message or System Message which does not have a valid value within the current database will be converted to Zero!

Notes:

In order to make the storing of process tables useful, it will be worth working out a standard of vocabulary, message, location and object allocation - perhaps based on your current adventure. E.g. If Sysmess 60 is always a message to prevent you putting something in a container. Then your tables can safely be transferred without modification, just with new text for the message in each game. This shell database would be standard for all games, merely assembling the required common PTF's for each new game, and typing in the message/object/location text and new vocabulary words.

If you make message 0 "Not implemented" or something similar. You will be able to detect errors because of message conversion after a PHOSIS table load. Note that if you seem to be getting a lot of "It's too dark to see anything" messages, or even the source of light (object 0) doing a lot, you probably have the same problem with the system messages or objects! If you end up at location Zero a lot you will be suffering from GOTO 0!

INFO

Format:
I [#stream]

Description:

Provides information on the screen (or to the optional stream) about various parameters of the current database. E.g. Number of

Locations, Messages, Objects, Process tables etc.
Note that I(nfo) is not stored for use by the R(ecall) command.

RECALL

Format:

=

Description:

Will provide the last entered line for possible amending -except I(nfo). See the notes under Transfer and Copy for its use in handling groups of entries. It can also be used usefully to reduce the typing for Verify, merely recall the Save command and change the S to a VI

Direction Pointer Table Amend

This would have been in PAW-TEL but there wasn't any room! It displays the 12 direction word values DPT, as used by PAW-TEL, for editing. Note that if there was no DPT in the database a null (all zeros) one will be inserted and presented for editing. See the PAW-TEL Map section for more details on the DPT.

TEL

PAW-TEL is designed to allow a database to be analysed and debugged with ease. It does this by providing a range of powerful functions to search the database under various criteria. The functions Hunt and Search allow entries to be found that use certain sequences of CondActs, or CondActs which refer to certain aspects of the database such as messages, objects etc respectively. This will allow all entries dealing with an aspect of a game being modified to be found more reliably, and easily, than by studying listings. The List function provides extensive features to identify currently used Flags, Objects and so on, to allow them to be reassigned reliably or new ones to be allocated.

Auxiliary menu

For all the following functions the Process table they are to use can be specified in one of three ways represented as [Process|Process, Process|?], thus:

Process

Specify a specific process table number.

Process,Process

Specifies a start and end process table number - note that the comma is required.

?

All process tables (i.e. from 0 to the last inserted).

It is possible to specify a stream as part of the Hunt, List and Search commands. This will then become the current stream used by commands as a default, you will not need to specify the stream specifically until you want to change the default again. The Info command will tell you which stream is current - on the screen!

The Info and Map commands provide the facility to output the data to a printer via a Copy function if it is implemented. Merely press key C while the 'Press any key' prompt is displayed. This will have no effect if the current printer driver does not support COPY.

MAP

Format:

M [Process|Process,Process|?] [locno]

Description:

Draws a pictorial representation of the exits from locno, giving the location numbers and exits from the surrounding locations. It uses information gleaned from the Connections table and entries in the given Process table(s). The exits drawn can represent the full 12 direction words; compass directions, up/down and

enter/leave. They are drawn as 12 radial lines from the mapped location. The meaning of each line as a word value is given by the Direction Pointer Table (DPT), it does not even need to be a direction word. There is a built in DPT in TEL which draws the directions as shown in Fig 1.

Connection table exits are indicated by solid lines. Map will find entries in the Process tables given which involve an AT locno1 GOTO locno2 sequence - although the two CondActs do not need to be immediately after each other. It will not include exits implied by range Conditions formed from ATLT, ATGT, NOTAT or SAME 38 flagno., NOTSAME 38 flagno. or combinations of same. Also Map will not find GOTOS in a sub-process called from an entry with an AT for the location being mapped. I.e. it will not delve beyond the Process entry being considered. Exits given by Process entries with a Verb that corresponds to a word in the DPT are drawn as dashed lines labelled with the Verb and Noun of the entry - although the Noun is usually * or . Where there is both a Connection table and Process table entry, for the same direction word value, you will get a solid line labelled with the Verb and Noun of the Process entry. Any Connection or Process entries that were found which did not have a Verb in the DPT are totalled and given as a count of Extra Connections on the bottom line along with the Process table range that was searched. Note: the Process entries could be found by doing a Hunt (see below) for AT locno. - where locno. is the mapped location!

Remember that pressing key C will COPY the displayed screen to a printer when the anykey prompt is displayed.

Direction Pointer Table

The built in DPT of TEL provides for the traditional directions, but, it is possible for each database to contain a DPT of its own using the Hunk allocation system of the latest versions of PAW. A redirection table for TEL can be inserted/edited in PHOSIS, this will then be used as a default by the Map function. It is possible to toggle between the Internal and Database DPTs as follows: Enter a Map command with a location of 255 (which can never exist), pressing ENTER will produce a flashing syntax marker after the 255, but the toggle will have been achieved. Merely delete the 255 and insert the real location number to Map.

In the DPT each entry contains the word value for the specific direction in the order:

N S E W NE NW SE SW UP DOWN ENTER LEAVE

The default table is thus:

5 2 3 4 9 6 7 8 11 10 12 13

Which if you take a look at the vocabulary are the word values of the directions given above! Although ENTER and LEAVE are not present in the default database.

The redefinition feature allows you to prevent certain directions being represented, by placing a zero in their respective entry in the DPT, perhaps to clean up a particularly cluttered display. Note that the DPT inserted by PHOSIS consists entirely of zeros. It is worth remembering that the entries thus excluded will be logged as extra connections. You can also use the DPT for databases which do not use the traditional directions or to customize the display for your own preferences.

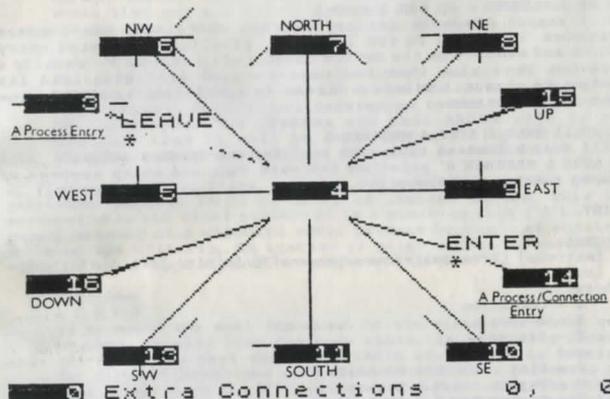


Fig 1.

HUNT

Format:
H [#stream] [Process|Process,Process?] [CondAct list]

Description:

Will search the specified process table or range of tables listing any entries, both as words and entry numbers, containing the given list of CondActs. Note that the CondAct list must be a

valid list with the correct number and ranges for the parameters. The matching is done absolutely so both the order and number of CondActs must be the same. It will though find the sequence anywhere within a larger sequence of an entry.

Examples:

H 7 AT 2 CARRIED 7

Will search all the Process tables (including Response which is Process 0) for entries containing the CondAct sequence 'AT 2 CARRIED 7'. Note that it will not find entries with 'CARRIED 7 AT 2' as the matching is done absolutely.

H #3 0 PRESENT 6 EQ 100 2 GOTO 2

Will search Response for any entries containing the CondAct sequence 'PRESENT 6 EQ 100 2 GOTO 2', printing the listed entry words and numbers on the device connected to stream 3 - usually a printer. Note also that the next command will also send its output to stream 3 unless a stream is specified, as this is now the current stream.

H 10,11 ATLT 7 ATGT 4 MESSAGE 8

Will search Process tables 10 & 11 for the CondAct sequence 'ATLT 7 ATGT 4 MESSAGE 8' printing the Verb Noun and entry numbers of those entries containing it.

LIST

Format:

L {#stream} [Process|Process,Process{?}] [C|A|E] [F|L|O|M|S|P|W] [no|word{?}]

Where:

C - Condition
A - Action
E - Either
and
F - Flag L - Location
O - Object M - Message
S - System Mes P - Process
W - Word

Certain combinations of options are not applicable and will be locked out by the syntax checker. They are C|E:M C|E:S C|E:P A|E:W - see below for details.

Description:

List is designed to do two essentially different jobs:

1/ Finding entries that refer to specific flags, objects, messages, words etc, without reference to the specific CondAct used. This is selected when the final parameter to the List is a vocab Word or Number.

Examples:

L 0 A P 10

Will list the Verb, Noun and entry number of all the entries in Response (Response=Process 0) which contain A(ctions)

that refer to Process table 10. In actual fact there is only one action (PROCESS) which fits this description. Thus A(ction) is the only valid first option - attempting to use C(ondition) or E(ither) will result in a syntax error.

L 2,7 E F 100

Will list the Verb, Noun and entry number of all the entries in process tables 2 to 7 which contain E(ither) Actions or Conditions which refer to F(lag) 100. This will include ZERO, NOTZERO, EQ, NOTEQ, GT, LT, SAME, NOTSAME, COPYOF, COPYFO, WEIGH, SET, CLEAR, PLUS, MINUS, LET, ADD, SUB, COPYFF, RANDOM, MOVE, WEIGHT, PRINT, RAMLOAD! This means it would list quite a lot of entries if flag 100 did much in the database - but, you will have found all its uses!

L ? C W BROWN

Will list the Verb, Noun and entry number of all the entries, in all the process tables in the current database, which contain Conditional references to the word BROWN of its word type. The chances are that BROWN will be an adjective thus it will be entries containing either the ADJECT1 or ADJECT2 Conditions that will be found.

2/ Finding out which flags, messages, objects etc are used in any particular process table or group of process tables. This is selected when the final parameter is a question mark '?' - the normal meaning of a wildcard would produce hundreds of entries matching the criteria. So instead it gives a list in ascending order of the flags, objects, messages etc in use.

Examples:

L ? E F ?

Is probably the most important of the variations which can be used. It will list for each table, in ascending order, all the flags that are used within it. I.e. it is looking for E(ither) Conditions or Actions which have a parameter of type F(lag) with Any ('?') value as a parameter. Rather than getting the entries, you get a list of all the values that the question mark could take - the flags used in the table. This of course allows you to quickly discover if a certain flag is used and if so where (by using Search to find uses of it) allowing you to decide if it can be safely reused and if not which flags are free!

L 3,10 A P ?

Will list in ascending order the (P)rocess tables, which are referred to in A(ctions) (the PROCESS action only actually), in Process tables 3 to 10. Thus you will be able to discover which tables use each other!

L ? A M ?

Is another useful one which will list, in ascending order, all the messages used in the Process tables. This would allow you to find redundant messages for reuse and so on.

SEARCH

Format:
S [#stream] [Process|Process,Process;?]
[CondAct[no|word]?[no|word|?]]

Description:

Will search the specified Process or range of Process tables for CondActs which match the CondAct specification on the command line. Note that Search can specify only a single CondAct unlike Hunt. This is because you can specify a question mark '?' instead of the parameter(s) of the CondAct if required. The question mark '?' is what is known as a 'wildcard', which simply means that the search will ignore the actual VALUE of the parameter.

Examples:

S 0 ISAT 7 ?

Will list to the current stream the Verb, Noun and entry number of all the entries in Response (Process 0) that contain ISAT CondActs which test for object 7 at any location number - this is indicated by the question mark which replaces the second parameter of the ISAT CondAct, which is usually the location number.

S ? RANDOM ?

Will list to the current stream the Verb, Noun and entry number of all entries in all the Process tables in the current database which randomly sets ANY flag.

S 3,30 SWAP 0 ?

Will list the Verb, Noun and entry number of any entries in Process tables 3 to 30 which swap object 0 (as the first parameter) with any other. Note that unless you are very consistent with your use of parameters to SWAP you will probably need to 'S 3,30 SWAP ? 0' to get all the SWAPS of object 0!

INFO

Format:
I

Description:

Provides information on the screen about various parameters of the current database. E.g. Number of Locations, Messages, Objects, Process tables etc. Also shows the current output stream number.

Note: The I(nfo) option on PHOSIS has a stream option, but TEL does not. This is because TEL keeps track of a current stream which would be destroyed by the I(nfo) to a stream, and one of its functions is to display the current stream! The display can be copied to a printer by pressing key C while the anykey prompt is displayed.

MEGA

MEGA is designed to reduce the amount of overlay swapping required for 48K Users of PAW - and 128K owners who are using 48K mode or have succeeded in using Page 7! It basically provides all the functions of Overlays 4 and 5 (the text and process editing menus) in one overlay, by shrinking and merging parts of the code and by omitting the Background Colours option. This allows messages, objects and locations to be inserted and edited, then the words and Process entries which deal with them to be created, without loading any overlays - except MEGA originally!

The sub-menu

MEGA's sub-menu looks remarkably like the main menu for PAW, we aren't about to repeat the PAW manual as you will find that all the functions work almost identically to their main menu counterparts. There are some slight differences though:

The options displayed can be split into four groups:

Internal

The Internal options are those which are handled within MEGA itself, i.e. Overlay 4 and 5 functions except Background Colours. The Save/Load/Verify Database options are also internal. When you use these options the Z function will be return to the MEGA sub-menu.

External

The external options in MEGA return to the main menu of PAW after usage, because they are usually in overlays which destroy MEGA! So you will have to reload MEGA when required. The ones to beware of are: Graphics, Default Colours, Compression, Test Game. Disc/Microdrive users should also be aware that Device Select is an overlay!

Free Memory

If you are using a cassette version of PAW then although Free Memory will return to the main menu, it does not destroy MEGA. So selecting user overlays (Z) and MEGA again (M) will not need to load the MEGA overlay. Disc versions DO hold Free Memory in an overlay so MEGA will be overwritten by the selection of it.

Return to Main Menu ('@')

This merely takes you back to the main menu of PAW.

The Response and Process tables are treated synonymously as described in the introduction.

PAW - PHOSIS/TEL/MEGA

User overlays for the:
Professional Adventure Writing System

KelSoft have produced some of the most consistently useful addon products to various adventure writing systems. We have pleasure in providing their three latest products, now converted to run as PAW User Overlays.

PHOSIS

Provides PAW with powerful Process table editing facilities. Move/Copy CondAct groups between entries or tables, Print and Delete selected entries and Save/Load entire Process tables between the same or different games. This allows a library of useful routines to be built up on tape or disc for use in later games.

TEL

Is a flexible PAW database analyser. Allowing the database to be searched for CondAct lists or CondActs with 'wildcard' parameters. TEL also provides functions to list usage of Flags objects, messages etc. according to various criteria. e.g. which tables to search, whether referenced as an Action, Condition or Either. The Map function provides a visual display of the exits from a given location.

MEGA

Reduces the overlay swapping required for 48k owners by providing all the Main Menu options of Process 4 & 5 in one load!

Program Design & Code: (c) 1987 G.T.Kellett T/A KelSoft
Published by: Gilsoft International Ltd, 2 Park Crescent
Barry, South Glamorgan, CF6 8HD.