

## PCW / CPC CPM Art

---

The graphics upgrade for the CPC/PCW allows any adventure you have written which has about 2.5K of TPA space available to have disc based graphics added. Note that the upgrade is not a graphics package, we assume you already have one of those. It will accept any screen format file you have. The screen runs in Mode 1 (40 columns, 4 colours) on the CPC so pictures must be in this format. Also note that the graphics are different for the CPC and PCW for obvious reasons!

The files supplied are:-

### CPC

MCPIC.COM	Make a CPC format PICCY file
ADDARTC.COM	Add or Update the graphics program to a PAWed game
ECPC.Z80	The graphics program
TICKET.MAP	An example of a .MAP file for the TICKET game

### PCW

MPPIC.COM	Make a PCW format PICCY file
ADDARTP.COM	Add or Update the graphics program to a PAWed game
EPCW.Z80	The graphics program
TICKET.MAP	An example of a .MAP file for the TICKET game

These files will be found on the respective sides of the disc in addition to the files described in the manuals with PAW.

### Overview

The system is designed to be as simple as possible to use. You can select to use any number of screen lines from 2 to 20 to contain the picture - the picture files will contain only the required amount of data to save disc space. The remainder of the screen is set up as a text window. You have the option of a blank line before the text window to separate the graphics and text. Additionally the CLS on describe can be disabled to provide a continuous scrolling window.

It patches into the CLS vector of a PAW game. That is to say it adds a new bit of program to the PAW game where it clears the screen! This extra code works as follows:-

On the CPC the screen is set to be MODE 1 (40 columns with 4 colours). And under CPM+ the status line is disabled. The text window is set to what remains of the lower screen area, after the number of specified lines are reserved for graphics and possibly a blank line.

When PAW attempts to CLS, just before describing a location, the new code is given control. If CLS on DESC is selected then the text area is cleared. Note that as this is carried out by the

interface code on the CPC, the CLS codes given using PAWINST are meaningless.

The colours specified by the .MAP file entry for this location are set up on the CPC.

If the game is in darkness then the area reserved for graphics is cleared and a return made to PAW. This is in effect a display of Picture 0, as this is considered as a clear screen.

Otherwise, if the picture specified in the .MAP file (see below) for the current location is not already on screen then the picture is loaded from the disc. If the picture can't be found a suitable message is output to the text window. This also happens if an error occurs while loading the picture.

Simple as 1.2.3.4.5 etc.

The steps to produce a graphics game for either CPC or PCW are:-

Write and test your game using PAW.

Save a standalone (.COM) version using: PAWINT name C.

Use a graphics package to draw some pictures.

Convert the pictures to PICCY files using MxPIC.COM

Write a .MAP file using a text editor to describe no of lines, colours and which location contains which picture.

Use the ADDARTx.COM program to add the graphics patch and .MAP files to the final game.

Use PAWINST to install the interpreter for the number of text lines and columns.

Play the game!

### Making suitable PICCY files

MCPIC picno file{.BIN} {lines}

Makes a file from a Mode 1 .BIN file (one which contains a 128 byte AMSDOS header) called PICCYxxx.CPC (where xxx = picno), containing 'lines' lines (default 12). The .BIN file is made thus under AMSDOS:

```
MODE 1
```

```
SAVE "screen",B,&C000,&4000
```

and is the format used by the majority of Art Programs available.

e.g.

MCPIC 2 falls

Will create a file PICCY002.CPC using 12 lines of FALLS.BIN

MCPIC 97 MYPIC.PIC 20

Will create a file PICCY097.CPC using 20 lines of MYPIC.PIC.

(n.b. it is still assumed that the file contains a 128 byte AMSDOS header even though the extension is different.)

MPPIC picno file{.PCP} {lines}

Makes a file from a screen image .PCP file called PICCYxxx.PCW (where xxx = picno) containing 'lines' lines (default 12). The file format of .PCP files is that used by MasterScan/Paint.

e.g.

MPPIC 2 falls

Will create a file PICCY002.PCW using 12 lines of FALLS.PCP.

MPPIC 97 MYPIC.PIC 20

Will create a file PICCY097.PCW using 20 lines of MYPIC.PIC.

The patched interpreters look for files in the format given above. Note that if you change the number of lines you use for a picture in the .MAP file, you will have to remake the pictures. This is because if you decide to use more lines, there will be insufficient data in the PICCY file and an error will occur while loading it. And conversely if you use fewer lines you will have too much data in the PICCY files thus wasting disc space.

### The .MAP file

This describes the format of the screen and describes which picture should be displayed at which location. It can be prepared in the same way as the .SCE file with any text editor.

Lets look at an imaginary example for TEWK. We will assume that we have three pictures, each of 12 lines:-

Picture 1 will be an introduction screen (used at location 0)

Picture 2 will be inside the ship (used at locations 1-3)

Picture 3 will be outside the ship (used at locations 4-5)

The .MAP file would be:-

```
;MAP for TEWK graphics
;
;Lines   Flags      Ink 0      Ink 1
12      1           0          26
;
;The links
;
;Loc Pic  Ink1  Ink2  Ink3
0  1    26   2    18      ;Could be any colours
1  2    26   3    4       ;Just because it's the same piccy
2  2    26  13   14      ;doesn't have to be the same colour
3  2    26  13   14      ;but it can be
4  3    26   4    5
5  3    26   4    5
;Done
```

Any blank lines will be ignored as will those starting with a semicolon ';' or an asterisk '\*'.

The first valid data line must contain the number of lines used

by each picture. The flags (see below) and the starting values for Ink 0 (Border and background) and Ink 1 (Text colour) in that order.

The flags are defined thus:-

- 0 - no blank line to separate picture and text. CLS on DESC
- 1 - blank line. CLS on DESC
- 2 - no blank line, Don't CLS on DESC
- 3 - blank line. Don't CLS on DESC

Although you can set other values - DON'T. They are reserved for future expansion and may have undefined effects.

The following lines then define any locations that contain graphics. In the order; location number (0-251), picture number to display (0-250), Inks 1 to 3 (0-26). One line is used for each location but they don't have to be in any order, and not all the locations need be specified.

e.g.

```
11 1 26 0 0
10 1 26 0 0
```

```
8 2 26 0 0
52 2 26 0 0
```

will define only locations 8,10,11 and 52. Any locations which don't receive a definition will default to the Global Ink 1 (CPC only) specified in the first line and picture 0. Picture 0 doesn't actually exist but is a Clear screen. This is also used for Darkness.

On the CPC the value for Ink 1 can be changed at every location if required. This should be undertaken with care. We suggest you attempt to draw your pictures with a fixed Ink 0 and 1 (the ones you are going to use for the game) and change only the values for Ink 2 and 3. If you need to change Ink 1, make it a small change such as sticking to bright or pastel colours that aren't too different. E.g. yellows and reds, or blues. This is because the text is always written in Ink 1 and it can be very distracting for the player to have this colour change continuously. This is especially true if you disable CLS on DESC, to create a continuous scrolling window, as the text on screen changes colour!

Note that the INK values will be ignored by ADDARTP, so you can use the same .MAP on CPC and PCW if you want. Although it seems unlikely that exactly the same pictures would be drawn for each machine.

If you define a location more than once then the last definition will take priority, no warning is given...

## Adding the graphics to a game

N.B. You can only add the graphics patch to a finished game. That is one that has been saved with the 'C' option of the interpreter as a self contained .COM file.

```
ADDARTx game{.COM} {mapfile{.MAP}}
where x = 'C' or 'P'.
```

This actually does the work of adding the graphics program/patch to the game and installing a machine readable copy of the .MAP file. The game defaults to an extension of .COM and if you omit the mapfile name it will default to the gamename with an extension of .MAP. If the game already contains the graphics patch then the program will update the code (in case of new versions) and change the .MAP file.

You must use the correct version of ADDART for the machine you intend the program to run on - although all the utilities supplied will run under any CPM. It's only the final program that is limited to one machine!. If you are making a version for both PCW and CPC remember to make a copy of the .COM file. The best method would be to have two .COM files for example TEWKC.COM and TEWKP.COM or similar.

e.g.

```
ADDARTC TEWK
```

Will add the files TEWK.MAP and ECPC.Z80 to the file TEWK.COM, changing the necessary code in TEWK.COM to patch the graphics program into the CLS vector.

```
ADDARTP TICKET BLANK.TXT
```

Will add the files BLANK.TXT (assumed to be a MAP file) and EPCW.Z80 to the file TICKET.COM.

## Installing the file

You must install the game.COM file to change the no of lines and columns that PAW thinks the screen has. This is done with the PAWINST program from the original PAW program disk.

For example on the CPC you have 25 lines. If you are using the default of 12 lines per picture and have set the flags to include a blank line then 12 lines remain for text. Additionally as the screen has been changed to MODE 1 we now have only 40 columns per line and this change must be made as well.

On the PCW from the original 32 lines you would have  $32-12-1 = 19$ . The screen remains with 90 columns so you shouldn't need to change that option.

If you don't change the installation then wordwrap and 'More...' will not function correctly.

As has been mentioned before the CLS is carried out within the graphics program on the CPC. So the installation option for this is redundant on programs which contain graphics.

## Running the final game

Once you have carried out all the above steps you are ready to try the program with graphics. This is carried out quite simply by typing the name of the program as before.

**NOTE:** The graphics patch will work on one machine only. Don't try to run a game which contains the CPC graphics patch on a PCW (or vice versa). If you are lucky it may crash...

If you haven't placed any PICCY files on the disk yet you will be getting 'Picture xxx not found' messages. All that remains is to add the pictures to the disc. Again you must place the correct pictures on disc, CPC files for the CPC etc...

It is possible that insufficient memory is available to run the program. In this case an error message will be produced and the game aborted.

## CPM 2.2 and CPM +

A game with the graphics patch installed when run under CPM+ may report a 'Insufficient common memory, remove RSX's' error. This will occur if too many resident programs such as monitors and RSX's have stolen memory above C000h. The only solution is to restart the CPM system without them present.

CPC Only:

All the programs provided will run under CPM2.2 or CPM+. Additionally because the CLS is now carried out by the graphics program, the same .COM file for your game will work under either operating system as long as it fits in the TPA... The problem occurs for those who intend distributing the final game, and indeed occurred with text only games. The TPA on CPM+ is much bigger, but if you want your game to run on the 464 with DDI or the 664 machines you must account for the fact that these machines only have CPM2.2 available.

The options are as follows:

The game must be written to fit in a TPA of 43K. As you need 2.5K for the graphics patch that means only 40K of TPA. If you compile the game within this size of TPA the compiler must say about 2600 bytes free for the graphics patch to be successfully added. If you compile in a larger TPA you can still ensure it will fit by checking that the end address, given by the compiler after compilation, does not exceed 41472 - but please try it in the smaller TPA before releasing your game.

Provide two versions of the .COM file both of which contain the graphics patch. One for CPM2.2 the other for CPM+. The CPM+ version can then contain more text, for example for the location descriptions. This is because with 20K more TPA you can go upto about 61440 and still fit the graphics patch on the end of the file! The pictures will be the same for both .COM files.

Note that due to the need to disable the terminal emulator of CPM+ on the CPC, and thus the status line, the 'disc not ready' error message will not be produced. Instead the graphics patch will report 'File not found' if there is no disc in the drive. Serious hardware errors may have unspecified results... CPM 2.2 will produce the 'Retry, Ignore, Cancel?' messages in the current text window as expected.

- - - - -

Programs and Documentation (c) 1990 Tim Gilberts. Thanks must go to Graeme Yeandle for the original work to develop this upgrade.

Distributed by:

GILSOFT, 2 Park Crescent, Barry, South Glamorgan. CF6 8HD