



Frank Neatherway

Before we begin the issue, I am extremely sorry to report that WACCI chairman Frank Neatherway passed away recently.

Frank was a great supporter of WACCI, as *Fair Comment* typist and then in a succession of committee posts. A fundamentally decent and honest man, he was generous in spirit and, though not hesitant to say what he thought was best, always managed to do so in an understated, friendly and helpful way. He will be sadly missed.

Many of you knew Frank much better than I did, and we would be pleased to print your memories of him in the next issue.

WACCI again

Yes, it is a (very much delayed) WACCI. Yes, it is smaller than you'd expect. Are you sitting comfortably? Then we'll begin...

It's become increasingly apparent over the past few years that WACCI is finding it difficult to survive in its historic form: a regular 32-page, A4 magazine about all things CPC, drawing on a regular pool of contributors. Membership levels have dropped as the CPC becomes purely the preserve of enthusiasts; contributors have become few and far between; editors have found it more difficult to spare the (huge) amount of time needed to put a magazine together. And there have been fewer takers for the essential jobs required to run the magazine and club, from editing *Fair Comment* to dealing with the membership admin.

Essentially, if we were to stick with the traditional format, there'd never be another WACCI. There wouldn't be enough articles to fill an issue, I wouldn't have the time to edit it, and no-one else has volunteered to help.

The good news

But in the fervent belief that a realistic (read: small) WACCI is better than none, we have a plan. WACCI will continue – but more like a newsletter than a magazine. Each issue will look like this one: A5 size, and around 16 pages long.

It's not fair to expect you to pay £2 for something this size, so we won't. WACCI is now formally run on a volunteer-only basis, so we only ask that you make a donation to

cover the costs of photocopying. Brian explains more in the letter you will have received with this issue.

You're also free to photocopy this issue for your friends, or, of course, to download it for free on the Internet. We hope to continue uploading the articles to the WACCI website, www.wacci.org.uk; but we'll also be making the magazine available as a downloadable PDF file. For those blissfully unaware of the intricacies of PC and Mac filetypes (good for you), this means you'll have one file on your hard disc, called *WACCI139.PDF*, which you can double-click - and WACCI opens up.

(I think this is a great thing, but it does mean that WACCI has taken a big leap backwards in presentation this time round - lots of text, not many pictures. It's now produced on AppleWorks under Mac OS X, because it can output PDFs; but I don't really know my way around it yet, and so have produced a fairly basic layout. It'll get better in the months ahead, I hope.)

WACCI will continue to be bi-monthly, and will hopefully stick to the timetable a bit more successfully than it has recently!

Coming soon

To which end, you'll be pleased to know that there are already a couple of articles ready for WACCI 140, including a hardware piece from Jonty, James Hoskisson's guide to fixing corrupt discs (thank you both), and a look back at those CPC programs that were much hyped but were never actually released. There'll be the answers to last issue's CPC Mastermind quiz, and maybe even another set of questions to answer.

As ever, your articles are very welcome, and so are your letters - hopefully, *Fair Comment* will make a return next time. Please send them to the usual Charlbury address.

Further reading

For those with access to the Internet, John Kavanagh's web-based CPC fanzine, *CPC Oxygen*, continues to prosper. John has recently embarked on the massive task of scanning and uploading all 119 issues of *Amstrad Action*, with the permission of Future Publishing. If you have web access, take a look at www.cpcoxygen.net. (And there'll be more on the CPC web browser in future issues of WACCI.)

Enjoy the issue.

Richard Fairhurst

Editorial: 5 Nine Acres Close, Charlbury, Oxfordshire OX7 3RD · wacci@systemeD.net
Administration: 'Number Six', Windmill Walk, Sutton, Ely CB6 2HN · 01353 777006

The A to Z of the CPC: Desktop Publishing

Richard Fairhurst takes a look back at
the history of CPC page layout programs

There are plenty of CPC entities with a good claim on the letter D. Discs. Databases. Demos. dkTronics. David Carter. David Long. Doug Webb. Danny Heatley. Derek Hyland. Ok, maybe not Derek Hyland.

But desktop publishing is a great example of the evolution of the CPC. DTP programs started with big-budget spectacles from Proper Software Houses, and developed through various stages of homebrew into an embarrassment of PD releases. (Or embarrassing PD releases, in several cases.)

And though this sounds like a case of “decline and fall”, it’s not. DTP programs were a great example of the CPC doing what it was good at. No-one pretended that a 6128 with an Epson LX-86 printer was going to produce professional quality results – but as time went on, the software became better suited to what people were using it for. To take two examples, you can’t disentangle Pagemaker Plus and PowerPage from the fanzines that were produced with them. Improvements in one prompted improvements in the other.

A mouse called Max

Though the CPC is usually bracketed with the Spectrum and the Commodore 64, the machine with which it had most in common technically was the BBC Micro: good BASIC, good OS, easily expandable, flexible screen display (neither the Spectrum or C64 had an 80-column mode). Both were equally suitable for games and serious uses, such as desktop publishing.

Neither the CPC nor the BBC could easily be hooked up to a laser printer, the miracle of technology that was driving the DTP boom on Apple Macs. But they had the CPU power and (most importantly) the disc drive for basic layout work. Fittingly, then, the CPC’s first two DTP programs were ports from the BBC.

One was called Pagemaker. Those with a grounding in computer history will recognise the name: Pagemaker was the Mac program that started it all, capable of conversing fluently in the PostScript language spoken by Apple’s LaserWriter and a million laser printers since.

This wasn't the same Pagemaker. The BBC and CPC ran AMX Pagemaker, the work of a Warrington start-up best known for the AMX Mouse. Someone, somewhere tipped off Aldus, the US authors of the real Pagemaker, who forced AMX to change the 8-bit product's name to Stop Press, previously the name of an AMX Pagemaker plug-in. (But keep the original name in mind. We'll return to it later.)

Today, it might seem odd for a US giant to pick on a CPC software company. It wasn't so weird back then. The 6128 was on sale in America, courtesy of local agents Indescomp; it was less powerful than a Mac, but not by much; and AMX Pagemaker promised to be rather good. Like Aldus's Mac program, it had windows, pointers, menus, fast scrolling... the lot.

AMX was never going to revolutionise the publishing world, largely because it worked in bitmaps (pictures made up of lots of dots) rather than vectors (pictures made up of resizeable and rescaleable 'objects' like lines, boxes and text). This limited output to the resolution of the CPC's screen, and made it difficult to correct your work without rubbing it out and starting again. Nonetheless, it was a very good start.

Fleet Street Editor was the other arrival from Beebworld. It was produced by the Daily Mirror, in its software house guise, Mirrorsoft. (Not as incongruous as it seems. British Telecom had a software arm, Firebird, and what was the BBC doing putting its name to a computer, anyway?)

FSE was a long time in coming, and performed well below expectations when it finally surfaced. Despite the quiet appearance of a bug-free, faster version 2 a year later, it never recovered from its initial panning by *Amstrad Action's* Pat McDonald.

Get out your cheque books

Meanwhile, a small Uttoxeter company called Creative Technology had brought out MicroDesign. The programmer was Simon Hargreaves, hence the 'Hogsoft' credit. MicroDesign had almost all the features you'd expect of a DTP program - large design area, drawing tools, clip art - and excelled at every single one. By employing the fantastic print quality of its 'strip mode', a skilled designer could produce better work with MicroDesign than with any other comparable program on the CPC.

But 'comparable', in many ways, referred to Advanced Art Studio rather than Stop Press. MicroDesign was great for posters and leaflets, but its rudimentary text-handling features ruled it out for serious DTP. There was no text layout feature of which to speak: you just typed onto the screen, and it was up to you to run the text around pictures or justify it by adding double spaces.

Still, some beautiful work was produced with MicroDesign. The first few issues of Tim Blackbond's *Artificial Intelligence*, for example; and scores of French fanzines, using

a translated version of the package called Oxford PAO ('publication assisté par ordinateur', I think). MicroDesign's A5 print-out wasn't quite full-page, so the French got into the habit of separately designing and printing an aside in a small box at the foot of each page.

A greatly expanded version, MicroDesign 2 (followed by 3), was released for the PCW. Its text-handling facilities were finally up to scratch, and output quality was breathtaking – even on a PCW dot-matrix printer. It deservedly saw off all-comers in the PCW market, including Stop Press, even though the latter's PCW version was a vast improvement on the CPC one (for starters, it didn't access the disc drive every five seconds). An unusual outfit, CT gave much of its profits to charity, but retreated from public view after over-extending itself as the lead developer on Amstrad's inglorious PCW16 project.

Put away your cheque books

There were a couple of home-brew commercial efforts, including Pagesetter by Martin Young and another from SD Microsystems. But the next revolution in CPC desktop publishing was avowedly amateur.

The PD boom was ignited by *Amstrad Action's* Free for All series and three pioneering libraries (David Wild's DW Software, WACCI under Steve Williams, and good old Robot PD). Soon, every 'squeaky' was starting their own PD library, the most successful of which was Alan Scully's Scull PD. As with most such libraries, it was loosely based on the Robot model, with faster delivery (not difficult) but less exciting new stuff.

Alan, myself, and several other PD librarians had the laudable habit of stocking up our libraries with our own home-grown BASIC programs. One of Alan's, which I can't believe took more than half an hour to write, was called Pagemaker. No relation to the Aldus or AMX programs, it was more a screen-design program than a desktop publisher. Working with a MODE 1, 40x25 screen, Pagemaker invited you to place any character you like, anywhere on the screen. This could either be text or one of the CPC's in-built graphic characters, allowing you to draw boxes.

So far, so simple. Pagemaker was less useful, less flexible than an art package, of which there were a couple in the public domain.

Life becoming a landslide

Mr Scully's next move was rather clever. Finding Pagemaker inadequate for his needs (namely the Scull PD catalogue, later to spawn the fanzine *CPC Domain*), he wrote an expanded version. Pagemaker Plus had the same basic principle as Pagemaker - a simple character-placement program. But the playpen had grown: where there was once a 40x25

screen, there was now a scrolling, 80-wide, 70-deep MODE 2 page. Perfect for printing out as A5.

Double-height text had been added to the original Pagemaker recipe, but otherwise, your pages were still restricted to the 224 letters and symbols of the CPC character set - though you could assemble some pretty blocky headlines this way. Pagemaker Plus deservedly caught people's eyes, including those of *Amstrad Action* and *ACU*.

I was a little intrigued by this, and carried out some back-of-the-envelope calculations. Because each PM+ character space, or 'cell', could only contain one textual character, it only took up one byte. So 80 columns by 70 rows would use 5600 bytes: 5.5k.

It would be much better to have fully bitmapped graphics, like a real art package, but 70 lines deep. This would allow clip art, mixing fonts, proportional text (the 'i's are slimmer than the 'm's), headlines of any size - you name it. A MODE 2 screen is 640 pixels across. A 70-line page, with each line 8 pixels high as usual, would be 560 pixels tall. $640 \times 560 = 358400$ pixels, which at 8 pixels per byte is 44800 bytes - just under 44k.

Let battle commence

It turned out that Alan and I were thinking along similar lines. His program came out first. The third in the Pagemaker series, Pagemaker Deluxe was 128k only, using the extra memory to store the 44k A5 page. It had full graphics editing, multi-size headlines, justified text - straight margins either side - and other doobries. And you had to pay £5 for it.

A remarkably similar program, Geoff Smith's Advanced DTP, was released at the same time, for the same price. Tim Blackbond reviewed both favourably in his new fanzine, *Artificial Intelligence*, with Deluxe marginally gaining the upper hand. PMD got a rave review in *Amstrad Action*, too.

Now, let's be honest, Alan and I weren't the best of mates. Scull and Robot baited each other in newsletters, fanzines, anywhere that'd listen - and several places that wouldn't. Blocs of allies shifted around the two camps, with the odd high-profile defection (David Carter, David Long) and double-crossing meat-brain (name withheld due to legal considerations, but it also begins with D). It probably looks very silly in retrospect, but for a bunch of 16-year olds, it was great fun. (Yes, we probably should have been out joyriding cars and graffiti-ing shopping precincts instead.)

So I was determined that PowerPage would beat PMD in every respect. It would be PD, not £5; it would have better text-handling, with pixel-perfect justification and columns flowing around pictures; it would import Stop Press .CUT clip art (which, admittedly, PMD also did, but only because Mr Scully had 'borrowed' my code) and export it, too; it would compress its saved files, to free up disc space.

Oh yes, and it would run in 64k. Alan refused to believe it could be done. But PowerPage did it, sneaking little fragments of machine code into the deepest crevices of the CPC's memory. Of the 44k page, 16k was held in screen memory at any one time, with the other 28k stored in the CPC's normal 42k program memory. Once input and output file buffers and fonts had been taken into account, there was about 6k left for the program. Eek.

AI liked it enough that Tim started to produce the whole fanzine on it. Many others followed. AA approved, too, featuring it on the covertape (you see, you don't have to charge £5 a copy to make money from something). New PD DTP programs continued to appear, but PowerPage was tops. Even if I do say so myself.

Return of the old stagers

Stop Press was a more powerful program, especially for A4 output, but many people found its menu system tedious and its constant disc access irritating. MicroDesign, though, was still as good a program as it had ever been.

Enter Kentish programmer Jess Harpur and Glaswegian entrepreneur Peter Campbell, known at the time as Campursoft (so imaginative) but subsequently to become Comsoft. Jess decided, with the connivance of Creative Technology, to give MicroDesign the text-handling features it had always lacked.

The result was MicroDesign Plus. The original MicroDesign program remained largely unchanged, but an 'extender' was bolted on to it, new features in a separately-loaded program. Of these, the most significant was a text import feature, allowing you to convert big slabs of text into clip art files ready for placement on the MicroDesign page.

The extender was WYSIWYG, of sorts: what you saved is what you got. This wasn't necessarily a great thing. If you wanted justified text, you had to save it as such in Protext; MD+ would then create a piece of clip art with exactly the same appearance as your Protext document. So the text would have been filled out to column width with extra full-size spaces in between each word; each letter would be the same width; and you couldn't flow the text around any graphics on the page. MD+ was still tops for graphics, unmatched for A4 output, but compromised for any sort of fanzine work.

And A5 fanzines were what was driving the CPC desktop publishing boom. You might remember some of the names: *Artificial Intelligence*, *CPC Undercover*, *Potential Difference*, *CPC Southwest*, *Presto News*. There were plenty more.

Nonetheless, MD+ did deservedly well, enough for Campursoft to think about a follow-up. Jess made a start on the project but never completed it, so Peter engaged Rob 'Radical Software' Buckley - author of *Fluff*, *Smart Plus*, and *Eve of Shadows* - to write it. It was never finished.

Meanwhile, back on the ranch...

...PD programs continued to appear. Alan Scully produced a cut-down version of Pagemaker Plus for the AA covertape (Pagemaker Plus-T) - in fact, the original could have sat happily on the tape without so much pruning. A short while later, PowerPage itself appeared on the tape, though it would only work once you'd unpacked it to disc.

A couple of the homebrew programs were eventually released into the public domain, and a few new projects were announced, such as David Long's As Easy As DTP, but never completed.

The original 64k PowerPage, meanwhile, was evolving into PowerPage 128. Conceived simply as a version of PowerPage without the need to load program segments from disc as and when they were selected, it grew and grew. There were proper pop-up menus, a zoom mode, and faster operation. Many features were requested by fanzine editors, whose publications were getting better-looking by the month. PP128 eventually settled down at version 1.2, though 1.3 is sitting half-completed on a disc somewhere.

Finis

And that was your lot. From mightily ambitious commercial packages (Stop Press) to simple PD releases (Pagemaker Plus) through to free-but-powerful programs (PowerPage 128), the story of desktop publishing is that of serious CPC software in a nutshell.

And as so often with serious CPC software, it all comes back to Protext. Yes, it's a word-processor. But it produced the first 50 or so issues of WACCI quite nicely, thanks to Promerge Plus's box mode, which allowed you to cut and paste columns individually - ideal for OMG's text-heavy style. Since then, various programmers have tried valiantly to graft on DTP capabilities (Prototype, Proprint, Pro-Ext), with mixed results. Sometimes it's best not to mess with perfection.

Programmers' Patch

Matthew Phillips delves into disc directories

As the Programmers' Patch mailbag has not even been underwhelmed with a letter from Mrs Trellis of North Wales, I am going to continue with the topic I was pursuing last issue (*I'm glad you didn't say 'month' there - Richard*). We had had a good look at how data is physically stored on a floppy disc. I perhaps should have pointed out that most of what I said about physical storage applies to almost any floppy disc on any computer. This month we are going to get on to stuff which is specific to the Amstrad (or rather to CP/M, whose disc format Amstrad adopted for Amsdos). I also forgot to point out the rather strange numbering of the sectors in that diagram of a disc which appeared. Sector C6 intervened between sectors C1 and C2. Peculiar. More on that later.

Right, now it is time to fire up your disc sector editor and start rooting around the Directory. Fire up Discedit, Sected, Discology, Xexor, DMon, or whatever and pop a typical Data disc in the drive. I am using Martin Schroeder's Discedit for the screenshots in this article. A listing for this was published in *Amstrad Action*, November 1987, but all sector editors look very similar when you get down to it, as there is not a lot you can do with an 80-column screen once you have displayed the sector data in hexadecimal and in characters.

```
Drv.:A   Pag.: 1   Tra.:00   ID: 01 C6 C2 C7 C3 C8 C4 C9 C5
3000  30 44 49 53 43 45 44 49 54 42 41 53 00 00 00 28  DISCEDITBAS 11C
3010  02 03 04 05 06 00 00 00 00 00 00 00 00 00 00  DISCEDITBIN 11C
3020  00 44 49 53 43 45 44 49 54 42 49 4E 00 00 00 03  DISCEDITBIN 11C
3030  07 00 00 00 00 00 00 00 00 00 00 00 00 00 00  DIR-SORTBAS 11C
3040  00 44 49 52 2D 53 4F 52 54 42 41 53 00 00 00 0E  DIR-SORTBAS 11C
3050  08 09 00 00 00 00 00 00 00 00 00 00 00 00 00  ← 11C
3060  00 41 52 54 49 43 4C 45 20 54 58 54 00 1C 00 07  ARTICLE.TXT 11C
3070  0A 00 00 00 00 00 00 00 00 00 00 00 00 00 00  DIR-SORTBAS 11C
3080  00 43 4F 50 59 44 49 53 43 42 41 53 00 00 00 0D  COPYDISCBAS 11C
3090  0C 0D 00 00 00 00 00 00 00 00 00 00 00 00 00  ↓ 11C
30A0  E5 53 46 4F 52 4D 41 54 20 42 41 53 00 00 00 09  *FORMAT BAS 11C
30B0  0E 0F 00 00 00 00 00 00 00 00 00 00 00 00 00  00 11C
30C0  00 44 45 50 52 4F 20 20 20 42 41 53 00 00 00 05  DEPRO BAS 11C
30D0  10 00 00 00 00 00 00 00 00 00 00 00 00 00 00  DEPRO BIN 11C
30E0  00 44 45 50 52 4F 30 20 20 42 49 4E 00 00 00 02  DEPRO BIN 11C
30F0  11 00 00 00 00 00 00 00 00 00 00 00 00 00 00  00 11C
```

```
↑ .. Cursor up   ↓ .. Cursor down   → .. Cursor right   ← .. Cursor left
e .. Hex/Ascii   j .. Next Page     / .. Write Sector  \ .. Track/Driv
f6 . Sector up  f4 . Sector down  f8 . Track up     f2 . Track down
```

Sector editing for beginners

In the main part of the screen there are three columns. The column of four-digit hex numbers on the left side is just for reference, so you can work out how far into the sector data each line is. It happens that Discedit starts the numbering at &23000 because that is the position it stores the sector data in memory. The middle area shows the bytes of the sector in hexadecimal. They are arranged in lines of sixteen bytes. Some sector editors put an extra gap after eight bytes to help you to count. On the right hand side, the appropriate characters from the Amstrad's character set are displayed, so it is easy to read off ASCII data such as DISCEDITBAS. Discedit displays all characters from 0 to 255: some sector editors just show a full stop in place of the control characters.

There are only 16 lines of data shown, and as most disc sectors on Amstrad discs are 512 bytes long, only half a sector fits on screen at once. With Discedit the ']' key allows you to flip the display between the two halves of the sector. Across the top the display shows which drive and track we are looking at, and then there is a list of the ID numbers of the sectors present on that track. In the screenshot C1 is highlighted because that is the sector in view in the main part of the window.

The data in view in the main part of the window can be edited, usually by typing in hex codes or by typing the characters. It is not actually written to the disc until you tell the editor to do so. Generally, any changes you have made will be forgotten when you move to another sector, unless you have not saved them to the disc.

The interleave principle

Again, in the list of sector ID numbers, you will see that C1 to C9 are jumbled up, with C6 intervening between C1 and C2. The track has been formatted this way in order to make it quicker to read the disc.

Often, when reading files from the disc, the computer will need to read several sectors in a row. After sector C1 has passed under the drive head, the computer takes a moment to digest the data it has just read, and there is a risk that the next sector on the track will have already started passing under the drive head when the drive is told to read the data. If this happens, the disc will have to make a complete revolution before we can begin reading again. This slows things down, so this trick of sector skewing is used. The next numbered sector on the track, C2, is actually the next but one physically, so we have plenty of time to digest the data without missing the start of the next sector.

Looking at the directory

With a Data format disc in the drive, move to sector C1 on track zero and you will see something similar to that displayed in the screenshot. You should recognize the names of

some of the files on your disc. On Data format discs the first four sectors are taken up with the directory. Each directory entry is 32 bytes long, recording both the name of the file and where it is stored on the disc. We will take a look at the first one as an example.

```

Row 1 0u f1 f2 f3 f4 f5 f6 f7 f8 t1 t2 t3 ex s1 s2 rc
      00 D I S C E D I T B A S 00 00 00 28
Row 2 d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 dA dB dC dD dE dF
      02 03 04 05 06 00 00 00 00 00 00 00 00 00 00 00

```

0u **user number**

&00-&0F: user number of file

&10-&1F: password (CP/M+ only)

&20 : disc name (CP/M+ only)

&21 : datestamps (CP/M+ only)

&E5 : deleted file or unused directory entry

f1-f8 **file name:** up to eight upper case ASCII characters, the rest filled with spaces

t1-t3 **extension:** up to three upper case ASCII characters, the rest filled with spaces

ex **extent number**

s1 **last record byte count** (CP/M+ only)

s2 **high byte of extent number** (if required)

rc **record count**

d0-dF **disc blocks allocated to this file**

These are the conventional names for each byte of the directory entry, with a brief explanation for reference. Let us look at them in more detail.

If you don't know what user numbers are... you haven't missed an awful lot. Look up the |USER command in your manual. The important thing to note is that when a file is deleted, most of the directory entry is retained: only the user number byte is changed to &E5. This makes it very easy to 'undelete' a file, providing that the space it occupied on the disc has not already been reused.

When a disc is first formatted, it is entirely filled with &E5, but CP/M does not bother to wipe the actual data from the disc when deleting files, as that would take time. Instead the directory entry is flagged in this manner, and that and the areas of the disc used for the file will be reused in due course. In the screenshot the sixth file, SFORMAT.BAS, has been deleted. CP/M+ discs can have other codes in the user number slot, and these are outlined in the table.

Names and extensions

The storage of the file name and extension (or file type) is easy to see, but note that the full stop that we are used to seeing between the name and the extension is not actually stored on the disc. The name must be stored in capital letters, or weird things start happening. If you fancy experimenting, then use your sector editor to change some of the names to lower case. You will find that the files appear in lower case when you catalogue the disc, but if you try to load any such files, Amsdos and CP/M will not find them, because the directory will be searched for the upper case name instead.

The characters for the name must be from the ASCII range 32 to 127, and many of the characters in that range are not allowed either. Amsdos, CP/M 2.2 and CP/M+ all have slightly different restrictions on which punctuation characters are permitted in file names. The general restriction that characters must be less than ASCII 128 means that the top bit of each of the bytes of the name and extension is potentially free for other purposes. Those in the extension, referred to as t1', t2' and t3', are used as follows:

- t1' read-only (file cannot be altered or deleted)
- t2' system (file is hidden and available to all users)
- t3' archive (file has been backed up)

You can try these out with your sector editor as well. For example, change the &42 at the start of the extension of DISCEDIT.BAS to &C2 and drop back to BASIC and catalogue the disc. You will find an asterisk appears by the file name, and you will not be able to delete the file using |ERA. The archive bit, t3', is intended to be used by software to make back-up copies of files, and is not otherwise useful. Under CP/M+, the top bits of f1 to f4 can be used for user (or application) defined attributes, but f5 to f8 must have the top bit clear.

The size of a cow

The rest of the bytes (ex, sr, s2, rc and do to dF) all deal with how large the file is and where it is stored on the disc. For Amsdos Data format discs it is quite easy to understand, but it can be much more complicated for larger disc formats. I will explain the terminology with the simple arrangement of Data format, and then we will see how it extends to Romdos and Parados formats.

The first concept to grasp is the "block". The whole of a CP/M disc is divided into equal sized blocks. On Data format discs they are 1K, which is actually the smallest block size CP/M allows. The first sector of the directory marks the beginning of block 0. On a Data format disc the sectors are 512 bytes big, so block 0 is made up of sectors C1 and C2

on track 0. Note that blocks can straddle two tracks. For example, block 4 is made up of sector C9 on track 0 and sector C1 on track 1.

The first four sectors of a Data format disc are taken up by the directory, equating to blocks 0 and 1. Thus the first block available for storing files is block 2. Have another look at the directory entry for DISCEDIT.BAS and you will find that the do byte has the value 2! Get the idea?

The second line of the directory entry (do to dF) tells us that DISCEDIT.BAS is stored in blocks 2, 3, 4, 5, and 6. Block 2 is made up of sectors C5 and C6 on track 0. Have a go at working out where one of the files on your disc starts, pop along with the sector editor and see if you can recognise it! For DISCEDIT.BAS, d5 to dF are all set to zero. The zero value means that there is no more space allocated to the file. Block zero itself is always the start of the directory, so it could never form part of a file's allocation anyway.

Precision engineering

Right, what does the record count (rc) indicate? At a fundamental level, CP/M works in "records" which are always 128 bytes long. That is because when CP/M was originally developed, floppy discs usually had sectors only 128 bytes in size. The rc value shows how long the file is in records. DISCEDIT.BAS has a record count of &28, or 40 in decimal, and 40 times 128 is 5120, or 5K. See how that matches up with the five 1K blocks allocated on the second line?

If you have a look at the other directory entries in the screen shot, you will see that it is not always this neat. DISCEDIT.BIN has a record count of 3, making only 384 bytes, much less than the 1K allocated. DEPROO.BIN, at the bottom, only occupies two records, or 256 bytes. This reflects the fact that there is a certain amount of space wasted on the disc, because files have to be made of whole 1K blocks. Nevertheless, CP/M records the length of the files in records, multiples of 128 bytes.

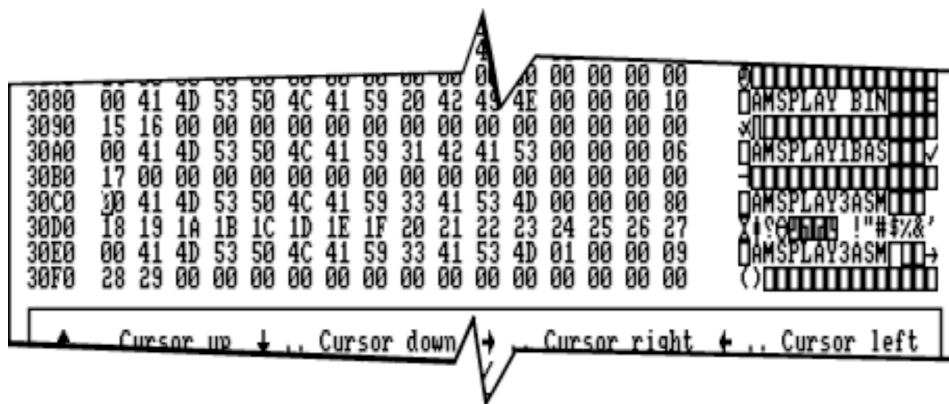
Most newer operating systems record the exact number of bytes in a file: Amsdos and CP/M 2.2 just round up to the next multiple of 128 bytes. As far as text files are concerned, the actual end of the file is usually signified with CTRL-Z (character 26, or &1A). Amsdos .BIN and .BAS files start with a header which tells the firmware the precise length of the file. For many other types of file, the fact that there may be a few bytes of garbage on the end does not actually matter.

When CP/M+ was introduced, a method of recording the exact length of the file was brought in. The si byte can be used to store how many bytes in the last record of the file are actually used. For the sake of compatibility, a value of 0 means that all of the 128 bytes are used. Very few CP/M+ programs actually make use of this feature, so you will rarely see anything but zero in this byte. In the screen shot you will see that ARTICLE.TXT has

si equal to iC and re equal to 7. This means that the file is precisely $6 \times 128 + \&1C$ bytes long, or 796 bytes. That was when I had only typed the first ten lines of this article! The si value was only set in this case because I had transferred the file from RISC OS using my own CPMFS filing system, which understands these things!

All creatures great (over 16k) and small

So what are ex and s2 for? If you have been thinking ahead, you will have seen that, as I have explained it so far, there is a problem storing files bigger than 16K. All of the do to dF values will be filled up, and there will be no way to indicate where the rest of the file is located. It's time for another screen shot.



This example is from side B of WACCI PD Disc 131. When you catalogue that disc, you will find a file `AMSPRAY3.ASM` which takes up 18K. As you see from the screen shot, there are two directory entries for this file. In the first one, blocks $\&18$ to $\&27$ are allocated to the file, and the record count is $\&80$, making 128×128 bytes, or 16K. In the second directory entry, blocks $\&28$ and $\&29$ are allocated, with a record count of nine, giving 1152 bytes, just over 1K. The important thing to note is the ex value in each case. The first directory entry has ex equal to zero, and in the second it is set to one. The chunk of a file recorded by one directory entry is known as an 'extent', and ex is the extent number. In this example `AMSPRAY3.BIN` consists of two extents.

One important thing to realise is that the extents of a file do not necessarily appear in the directory in the right order. Files created using Amsdos cannot be lengthened without saving the whole file again, but it is possible to lengthen a file using CP/M. If a new extent is started, the operating system will just take the first free directory entry it comes across. If a previous file has been deleted in the meantime, then that directory entry will be used, and the extents will not be in numeric order in the directory. Even more

common is a situation which can occur under Amsdos as well – where the directory entries may be in the right order, but are not consecutive.

The same applies to the blocks allocated to a file. The examples in the screenshots tend to be neatly arranged with consecutive blocks allocated to a file, but once you have deleted a few files and saved a few more, things will be very different. Take a look at a well-used disc in your sector editor, and you will see what I mean.

CP/M imposes no restrictions whatsoever on the use of the disc's storage. This might seem an advantage especially when compared with disc operating systems on other home computers of the same era. The BBC Micro, for example, had a system where files could only be stored in continuous blocks. After deleting a few files, you might have 20K free on the disc; but if the spare space was not all gathered together, it would not be possible to save a 20K file without using the *COMPACT command, which took ages – and might not be available if you were trying to save a file from an application.

There are, however, some disadvantages to CP/M's flexibility. If files are split up in little bits across the disc, it can be much slower to load them, as the drive head has to seek from one track to another much more. Also, it is harder to recover the files if you accidentally wipe your directory with your sector editor!

Keep it simple

Since the days of CP/M, operating system designers have continued with these different approaches. MS-DOS and Windows have the same flexibility and failings of CP/M: it sometimes pays to 'defragment' your hard drive to sort out the mess. Other operating systems (including RISC OS, the successor to the BBC Micro) have relied on more sophisticated algorithms to ensure that files are kept in one piece so far as is possible. Most systems split up the jobs of recording the names of the files and the spaces on the disc which they occupy. Meanwhile, CP/M's simplicity is a great advantage when it comes to delving in there with a trusty sector editor!

Next issue, we will look some more at the directory entry, and deal with the more complicated situation found on larger format discs where the capacity exceeds 256K. In the meantime, as a taster, I will leave you with a little BASIC program which works out from the block number what the corresponding track and sector numbers are. You can use it together with your sector editor, if you have two computers! It is set up for Data format discs, but if you alter the first two lines to set `lows%` to `&41` and `res%` to `2`, then it will give you the correct answers for System format discs instead. By the way, if you miss out all the percentage signs it will work just as well and will be quicker to type in.

The BASIC program

```
10  lows%=&C1
20  res%=0
30  spt%=9
40  sides%=1
50  tracks%=40
60  interleaved%=-1
70  bls%=1024
80  blm%=(bls%/128)-1
90  bsh%=(LOG(bls%)/LOG(2))-7
100 exm%=0
110 sectsiz%=512
120 psh%=(LOG(sectsiz%)/LOG(2))-7
130 WHILE -1
140 INPUT"Block number:",blk%
150 sector%=res%*spt%+blk%*(2^(bsh%-psh%))
160 FOR i%=0 TO (2^(bsh%-psh%))-1
170 t%=(sector%+i%)\spt%
180 s%=((sector%+i%) MOD spt%)+lows%
190 side%=0
200 IF sides%=2 AND interleaved% THEN side%=t% MOD 2:t%=t%\2
210 IF sides%=2 AND NOT interleaved% AND t%>=tracks%
    THEN side%=1:t%=2*tracks%-t%-1
220 PRINT"Side: ";side%;" Track: ";t%;" Sector: &";HEX$(s%,2)
230 NEXT
240 WEND
```


Meet the Relatives

Kevin Thacker investigates the CPC's mysterious East German cousin, the KC Compact

The KC Compact is a clone of the Amstrad CPC. It was made in 1989 by VEB Elektronik to celebrate 40 years of the German Democratic Republic. Soon after the Berlin Wall was taken down, east merged with west, and Germany as we know it today was born. However, production of the KC Compact ceased.

The computer is rare, and can fetch a high price on Ebay or other auction websites. I've seen them go for as much as £200. Initially I was part of a bidding war and lost, but then I was lucky, and found a person willing to sell his KC Compact. He accepted my more reasonable offer, and I imported it into the UK.

Since then I have discovered more about it, and I will tell you what I have found to date.

From the outside

Visually, the KC Compact is a cuboid shape (technical speak for a rectangular slab). The unit stands approx 2.5cm, similar to a CPC6128, is not as wide, but approx 50% bigger in depth.

The case is the same design as the BIC A5105, a educational computer also made by VEB with the same colour as the CPC6128+. The keys are flat and not as tall as the CPC keys, but have a better feel. The arrangement, of course, is different and the F5-F9 keys are missing. The Return key is below the cursors, and takes some time to get used to.

The connectors are also different. Looking at the back, from left to right come: Power-DC (a 'Telefunktion' connector), Tape (a 5-pin DIN socket, same as the CPC), UHF (a aerial lead which plugs into a TV), TV-RGB (a SCART socket), Printer (a 25-way D-type connector, same as the CPC+) and Expansion (a 50-way PCB connector).

On the right, from left to right there is: a power switch, JOYST (a 9-pin d-type; same as CPC joystick) and SOUND (a 5-pin DIN which can be connected to some old-style stereos/amplifiers).

The aerial and SCART work well with modern TVs, which can handle the different PAL signal encoding used by German TVs. If you're unlucky you won't get a stable picture on an older British PAL TV, or there will not be any colour. (However, I have been successful connecting the KC Compact to a CPC+ monitor.)

I tried two modern TVs, a 14-inch Sony portable and a large 22-inch Panasonic. The picture on the portable was visible but blurred – the TV definitely had a problem with the signal. I tried connecting the SCART, and for a few seconds, the picture was perfect. Then it shifted and it was difficult to see what I was typing. In contrast, the Panasonic didn't have a problem. The picture was perfect using the aerial and SCART socket.

The cassette socket on the KC Compact is identical to the CPC's, and I used my old cassette lead with great success. I've not tried the other connectors yet, because I need to make some conversion leads.

The KC Compact requires a separate power supply block to convert mains voltage (approx 230V AC at 50Hz) into the strange -20V DC and 20V DC required by the computer. Internally the computer has a voltage regulator which converts this signal into -12V, +12V, -5V and 5V. These voltages are also available at the expansion connector. (I'll have to be careful when I make my conversion lead, because I don't really want to fry any of my CPC peripherals!)

From the inside

So much for the external differences. Let's move onto the software and hardware differences.

When the computer is switched on, the famous yellow on blue screen appears – but with the following, rather different, text.

```
KC Compact  
Version 1.3
```

```
BASIC 1.1
```

```
Ready
```

And yes, it is the same Locomotive BASIC v1.1 as used in the CPC6128. So CPC BASIC programs will work without any changes.

In fact, the operating system is also from a CPC6128, but with some extra patches added:

1. the start-up message has been changed;
2. there is extra setup code to initialise the hardware of the KC Compact; and
3. there is a simple communications interface built in.

Although the computer has the ROMs from a CPC6128, it only has 64K of RAM.

The simple communications interface allows a 256-byte sized program to be downloaded into the KC Compact memory. A lead attached to the expansion connector is then hooked up to another computer, so the two machines can communicate. I guess that this feature was used to test the computer during development, but I can think of other uses too – for example, it could be used for ‘cross-development’. (This is a method to develop a program on one computer and test it on another. The program is sent from the development machine through a cable to the destination machine, and run there. Because you don’t have to keep swapping between source code and the resulting program on the same machine, this makes program development much quicker.)

The extra setup code initialises the Zilog Z8536 counter and timer IC. This IC is used to generate the interrupts, and is also connected to the printer connector. In the CPC design, the interrupts are generated by the Amstrad-designed Gate Array: the KC Compact doesn't have a Gate Array, and the functions of this chip are emulated using lots of TTL logic chips. (The ICs inside the computer are mostly Russian knock-offs of familiar chips. For instance, the U880 is the Russian equivalent of the Z80 CPU.)

Compatibility

The cassette is entirely compatible with the Amstrad, and normal CPC programs can be loaded using it. However, there may be problems running some games. If a game relies on the exact interrupt generation mechanism of the CPC, then these are likely to fail. I haven't tested any demos yet, but I expect there will be serious problems! (Demos often don't work on different models of the CPC, because of the different 6845 CRTIC ICs used. But the KC Compact has different interrupt generation and possibly different instruction timings, so could cause more problems.)

However, I still believe that the KC Compact is 95% compatible with the CPC. In some respects, it is actually more powerful:

1. The interrupts can be re-programmed. They could be set up to trigger every screen line, trigger once, or at any rate. This would allow interrupt-driven sample sound to be played very easily, something which requires painstaking CPU instruction timing on the CPC.
2. Also, since the Z8536 is connected to the printer port and is a I/O chip, we can use this as a general purpose 8-bit input/output port. This is far superior to the write-only 7-bit printer port of a standard CPC! This port could be used to transfer data between machines more quickly, to control another device, or for many other purposes.
3. Since there is a internal power regulator generating 12V and 5V, it is possible to connect external devices without needing additional power supplies.

4. From the schematics, I can see that the Z8536 is also connected to other parts of the KC Compact. I haven't found out yet whether it can be used to reprogram the video hardware, or to speed up the computer by reducing wait states.
5. The video hardware uses a ROM for the colour-palette lookup. This allows 4 different levels of red, green and blue. Therefore it might be possible to have a palette of 64 colours.

Some of these possibilities are speculation, based on observations of the schematics, and will only be proven when I have the time to try them out.

Clone king

VEB's cloning project didn't stop with the CPC hardware. They also cloned some software and peripherals.

Fifteen games packs were released (SpeileBox 1-15). Some contained software written by VEB, usually in BASIC; others contained patched games. These include modified versions of Speed King by Mastertronic and Arkanoid by Imagine/Ocean.

VEB also released some utilities, including a assembler/monitor program called ASMMON, which was actually Hisoft's Devpac! The amount of KC Compact-specific software is limited: I suspect most of it doesn't use the Z8536, so it should also run on a standard CPC.

VEB also made their own disc interface. The interface contained the same disc controller as the CPC, but also their own 64k memory expansion, which is *nearly* compatible with the CPC 6128's extra memory and dk'Tronics' RAM expansions. This interface was connected to a 5.25" disc drive. The disc operating system, again, was a patched version of AMSDOS, called BASDOS.

However, when it came to CP/M, VEB chose instead to use MicroDOS, a clone which also ran on their existing KC85/3 and KC85/4 computers. (These computers were not compatible with the CPC.)

Investigations are continuing

This is what I have found to date from various sources. I'm continuing to gather information about the KC Compact: if you have Internet access, you can read the latest information on the Unofficial Amstrad WWW Resource, <http://andercheran.aiind.upv.es/~amstrad/>.

Do you know any more? If so, please contact me. All contributions are welcome, and you will be credited.