

# Group Survey - Compression Systems

Richard Hallas performs a comparative test of the various RISC OS data compression utilities

When purchasing a hard disc, so the saying goes, you are usually told to think of the largest capacity you will ever need and then double it. It is a fact of life that you can never have enough storage space, either for physical objects or for computer data. If someone tried to sell you an appliance which claimed to turn your home into a Tardis, allowing you to store twice as many objects as there was space in the house to keep them, you would rightly be sceptical. However, in computer terms such claims may be valid, and adverts for utilities which profess to double the storage space on your discs can be very alluring. This survey aims to show the relative merits of the compression systems available for RISC OS machines.

Archimedes compression utilities have come a long way over the last couple of years. They started off by being capable only of compressing files for storage: you could keep files in a compressed form (an archive), but in order to use them you would have to decompress them first. Subsequently, the utilities took the process a stage further by performing the compression and decompression transparently, allowing you to save and load files directly from the archive, and the latest programs can integrate totally with all the other filing systems.

## DIFFERENT APPROACHES

There are currently six compression utilities in common use on RISC OS, and they fall conveniently into three pairs in terms of the way they work. Squash and PackDir are of the old school, in that they allow you to

compress files for storage but do not allow you to run the compressed files directly from the archive. ArcFS and SparkFS create archives which resemble normal directories (although they are in reality single files), and into which you can save and load files as normal. Under RISC OS 3, such archives behave just like any other directory (although their use is a little more complex under RISC OS 2). KleinFS and CFS take a different approach: they duplicate your normal filing system icons on the icon bar, and any file saved into a directory opened from one of these icons will be individually compressed. Viewed in a normal directory, the files have squashed icons, but in a compression directory the file icons appear as normal. The latter four utilities mentioned are all true filing systems, hence the FS suffixes.

## L-Z-WHAT?

Although they present themselves in different ways, the RISC OS compression utilities all use the same basic method for data compression: LZW. This algorithm (named after its creators, Lempel, Ziv and Welch) basically works by encoding frequently used sequences of bytes. The number of different sequences that can be encoded depends on the number of bits used for the codes, with greater numbers of bits resulting in better compression but requiring more memory for the compression process. The usual number of bits to be used for LZW compression is 12, but up to 16 is common. All the utilities tested here use 12-bit LZW compression, though some can give up to 16-bit LZW.

## SPEED, SECURITY AND SPACE

By their very nature, compressed files are always more susceptible to damage than uncompressed ones. Whilst a single disc error in the middle of a text file may (at its mildest) result in a spelling mistake, a disc error within a compressed file may result in the loss of the whole file. With a system like PackDir or ArcFS, using an archive file which itself contains a directory structure full of files, a single error could potentially result in the loss of all the files in the archive (although the programs employ measures to make this unlikely). There is also a reduction of speed in accessing your files, as all compression systems must take time to compress and decompress the files.

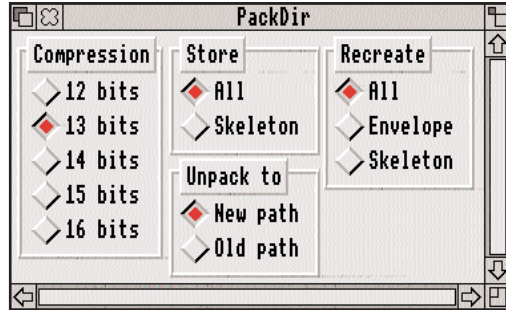
Squash is supplied free with RISC OS 3 upgrades and all new Acorn machines; the part of the utility which does the compression is actually built into the operating system.

Squash is very simple to operate, and creates compressed versions of whatever files you drag to the icon bar. Compressed files must first be

decompressed (by double-clicking on them) before they can be used. Rather than creating a single archive containing a selection of files, it compresses each file individually within a standard directory structure instead. As a simple utility, its performance is quick and straightforward, and only the standard 12-bit LZW compression is offered. However, any application

PackDir was created with the specific purpose of backing up discs in mind, and has been tailored for those needs. It will not accept single files for compression, but instead expects you to give it whole applications or directories, the contents of which it then saves out in a single archive which can also be read by SparkFS.

PackDir is extremely fast and efficient, and offers 12- to 16-bit LZW compression. It is by far the fastest utility on test here, although as it is not a filing system it does not suffer from the filing system overheads which should be taken into account. Although only really of



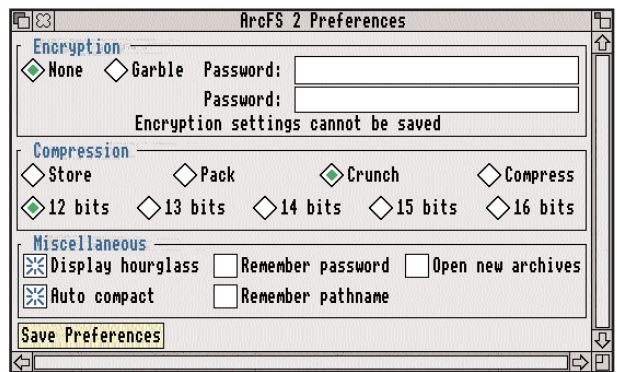
PackDir

relevance for backup purposes, it is very easy to use and extremely reliable. It is also very good for

ArcFS has been around for some time now, and is well into version 2. It has proved very popular, and is used by several magazines to compress their cover discs. Like PackDir, it offers 12- to 16-bit LZW compression. To use ArcFS, you must drag an archive file (which looks similar to a directory) into a normal directory window. Although this seems like saving a file, it is best to think of it as being like creating a new directory from a file menu.

The archive behaves in just the same way as a standard directory, and you can save and load files (and create sub-directories) in it. ArcFS's menu very conveniently lists all the archives which are currently in use, and lets you open their root directories or close them. Operation of ArcFS is speedy and convenient. Its worst current drawback is the fact that it uses memory in the Module area for compression. This is rather inefficient, and can lead to wasted memory which cannot always be reclaimed. However, the current

version of the software has provision for using a different area of memory and, assuming it is implemented in a future version, this would remove the



software's most annoying feature. ArcFS allows you to encode data stored in an archive, so that a password is required to read the data back. This is a good security feature, though it must be used with care; forget the

ArcFS

KleinFS makes use of RISC OS 3's Squash module for its compression system, and so is for RISC OS 3 users only. If you have been using Squash to compress files, then obtaining KleinFS will let you use them just as you would any other files on a disc.

Running KleinFS installs a floppy disc icon on the icon bar, and clicking on this opens a directory onto the floppy disc in drive 0. Saving files into this directory will compress them transparently, and of course retrieving files from it will decompress them as they load. It is just like using the disc as normal, except that files take up less space than they would otherwise. You can easily add a KleinFS icon for every drive you possess, and, usefully, you can give every single drive a custom

icon. Another extremely useful feature provided is the facility to kill off the standard Filer icons (you can revive them very easily). Killing the duplicate icons removes the possibility of accidentally opening a standard directory onto a disc when you meant to open a compressed one, and also makes it less likely for you to have two windows open on the same directory, which can be very confusing.

It has to be said that, overall, KleinFS is the slowest system under test here. However, it is extremely easy to use, and far less confusing in operation than its main competitor, CFS, so the advantages of using it could well outweigh the small speed loss for many users. It is also very good value for money, and

Compression (CFS) works in a similar way to KleinFS, installing drive icons on the icon bar. Unfortunately these all use exactly the same very dull sprite, so it is difficult to distinguish between them; also, it is not possible to disable the standard drive icons, so it's all too easy to click on the wrong one by accident. This can all lead to quite a lot of confusion for the user, and

especially for the novice.

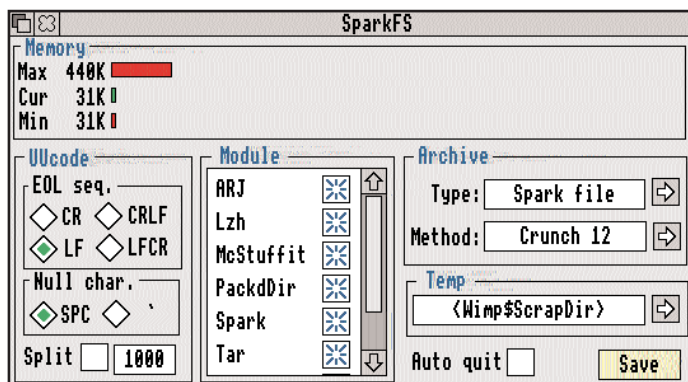
These criticisms apart, though, CFS works extremely well. Aside from PackDir (which is not a true filing system), CFS is the fastest utility on offer (for compression at least), and its adaptive LZW compression gives slightly better than average results.

SparkFS is the filing system equivalent of the very first RISC OS compression utility, Spark (which is still supplied on the SparkFS disc). Spark was able to cope with many kinds of archive, and this is equally true of SparkFS, which presents a very impressive array of options. It is entirely modular, so you only need to load the particular compression modules you need to use. If you never need to deal with McStuffit (Macintosh) archives, for example, you can forget about the McStuffit module. This approach enables SparkFS to deal with many kinds of archive without wasting lots of memory; and as new compression systems appear, it will be possible to extend SparkFS with appropriate new modules. Eight basic compression modules are currently provided with SparkFS, enabling it to deal with all the common archive formats on RISC OS, PC, UNIX and the Macintosh. SparkFS is ideally suited to the field of communications where different kinds of archives (which are incompatible with other RISC OS systems) are in common use.

When judged on the same grounds as other RISC OS compression systems, SparkFS is marginally slower than some of its competitors. However, it does not attempt to be the fastest utility; rather, it is the most comprehensive.

Product	Inclusive Price	Compression Method
Squash 0.30	Free	LZW 12-bit
PackDir 1.35	Free	LZW 12-bit <sup>†</sup> LZW 16-bit
ArcFS 2.23b	£31.36	LZW 12-bit <sup>†</sup> LZW 16-bit
SparkFS 1.26 <sup>2</sup>	£23.95	LZW 12-bit Zip Deflation
KleinFS 1.17	£17.95	LZW 12-bit

Notes  
Timings are in seconds  
All tests were performed on a machine fitted with a graphics enhancer board, and will correspond to the time taken in mode 0 on a standard machine.



SparkFS

Despite its wide range of options, SparkFS is extremely simple to use, and its approach is most

similar to ArcFS. Usually though, it is able to cross over the boundary into CFS- or Squash-type operation by use of directory-style archives, where files are compressed inside a standard directory structure. Conversion of archives between the two types is straightforward.

Although not the fastest, SparkFS is by far the most comprehensive and flexible system available. It performs all its own memory management efficiently, and is extremely well implemented. Its low price also makes it exceptional value for money.

A small point against SparkFS is that the only decompression utility supplied with it which can be

## TABLE NOTES

The table shows the results of five tests performed on each utility. Each test involved finding the time taken to compress and decompress a file or directory, and the table shows both of these times for each test in addition to the compressed size and percentage of the uncompressed original size. The application used in the test was a heavily customised copy of Impression Style, containing several extra files; the Directory contained twenty Draw, ArtWorks, text and Impression

(directory-style) documents; the Sprite was a mode 21 desktop screenshot; the Draw file was a complex piece of monochrome clip-art; and the Text file was an input file for the PMS music typesetting program, resembling programming source code.

12-bit LZW compression was performed by each program in order to give a fair comparison. In addition, however, those programs which had alternative methods on offer (PackDir, ArcFS and SparkFS) were

Application (1021K)			Directory (475K)			Sprite file (320K)			Draw file (16K)			Text file (58K)		
Comp. Time	Decomp. Time	% Size	Comp. Time	Decomp. Time	% Size	Comp. Time	Decomp. Time	% Size	Comp. Time	Decomp. Time	% Size	Comp. Time	Decomp. Time	% Size
31.5	17.5	70%	26	9	40%	4.5	1.5	13%	1.5	.75	55%	1.5	1	45%
19.5	18	70%	13	9.5	45%	(2) <sup>1</sup>	(1.5)	(13%)	(1.5)	(.75)	(53%)	(2.5)	(1)	(44%)
40.5	18	69%	20	14	45%	(3)	(1.5)	(12%)	(2)	(1)	(53%)	(2.5)	(1)	(39%)
93	26	71%	91	24	46%	4.75	3	13%	3.25	1.5	55%	3.25	1.75	43%
122	31	69%	115	29	46%	6.5	3.5	11%	5	2	53%	5.75	2.25	39%
94	32	71%	92	32	46%	7.5	4.25	13%	4	1.75	53%	4.75	2.5	44%
213	35	57%	225	30	35%	75	3.5	9%	5.5	1.5	40%	7.75	2	25%
119	43	75%	100	41	48%	15.5	4.5	13%	5.5	1.75	55%	7.25	2.25	45%
CFS 1.17	£45.82	LZW adaptive	40	48.25	71%	33.75	41.5	45%	5.25	3.75	13%			
2.5	1.5	51%	3.25	2.5	44%									

<sup>1</sup> PackDir and ArcFS can perform 12- to 16-bit compression

## CONCLUSIONS

All the programs on test here work extremely well and do generally live up to the claims of halving the size of your files on average. Obviously some files compress much better than others: fonts hardly compress at all (and attempting to run fonts from a compressed form is inadvisable due to the speed penalties), whereas sprites often compress to an extremely compact size.

Squash, provided with RISC OS 3, has its uses, and is a valuable operating system resource. However, for general archiving, one of the other systems should be considered. PackDir is very fast and efficient for its stated purposes, but is not intended for everyday use.

If you wish to compress all your files without having to think about doing so manually in order to increase the capacity of your hard disc, then your choice is between

CFS and KleinFS. Although CFS is very much the faster of the two, it is expensive and rather unfriendly to use.

KleinFS is cheap and exceptionally easy to use, and makes the dream of replacing your drive icons with compressing versions a reality.

If you wish to be in full control of what you compress, and not suffer from wasteful disc directory structures, then you must choose between ArcFS or SparkFS. Neither is suitable for compressing a whole hard disc in the way that CFS and KleinFS are, but they are better for general backing up. ArcFS is the faster of the two, and I have personally been using it constantly for the last couple of years. If you only need to access RISC OS archives, then it may be the best

