

Gerald's Column *by Gerald Fitton*

This month I'm going to tell you about a feature of PipeDream called Command Files. These were first introduced in PipeDream 3—but first a little history.

PipeDream 1

PipeDream 1 is an upgraded version of View Professional.

It exists in two forms. One of these is on the Z88 'LapTop' (I still have a couple of these early machines, one with a hard cover). The other goes by the name of PC PipeDream. It runs under MS-DOS as a single tasking program. I still get correspondence about the Z88 machine and the PC PipeDream program. When I get such correspondence I reply that spares for the Z88 and support for PC PipeDream are available from:

Vic Gerhardi
Rakewell Ltd
24 Putnams Drive
Aston Clinton
Aylesbury
Bucks
HP22 5HH

email: vic@rakewell.com and rakewell@nildram.co.uk
website: <http://www.rakewell.com/>

Telephone: (+44)1296 630617
Fax: (+44)1296 632491

The last time I checked the price of PC PipeDream was £58.75

PipeDream 2

PipeDream 2 was the first version of PipeDream released for the Archimedes. Like PC PipeDream, it is single tasking and runs under the single tasking Arthur operating system (sometimes designated RISC OS 1). When RISC OS 2, the multitasking version of RISC OS, was released, PipeDream 2 became obsolete almost overnight.

PipeDream 3

PipeDream 3 was a big advance on PipeDream 2. It does not have charts built into it but it does include a 'hot link' to a graphics package. This (undocumented) 'hot link' is still retained within PipeDream 4 and PipeDream 4.5. It can be used to run BASIC programs from within PipeDream and return values to PipeDream. This hot link is not my theme for this month. This month's theme is PipeDream Command Files. In PipeDream 3 these are called called Macros, their FileType is PDMacro and the icon contains a (red) M.



PipeDream File



Macro File

With a few exceptions, all Macro Files created using PipeDream 3 will run in PipeDream 4 and PipeDream 4.5. I still receive correspondence from users who regularly use PipeDream 3 rather than PipeDream 4 or 4.5. If you have only PipeDream 3 and have never used PipeDream 4 then you may not recognise “Command File”. The Command Files of PipeDream 4 with FileType PDCmdFil are the same as Macros of PipeDream 3.

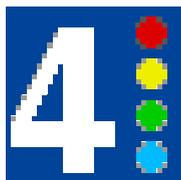
PipeDream 4

I suppose that the biggest single improvement introduced in PipeDream 4 was charts. At the time it was heralded as “One of the best charting packages around!”. However, this article is not about PipeDream’s charting package, excellent though it is, this article is about PipeDream Command Files—so charts will have to wait.

The biggest single complaint about PipeDream 4 was that many of the keyboard shortcuts were changed or removed; I prefer the expression ‘tidied up’. It is not true that many PipeDream 3 commands were removed completely; they were ‘hidden’. These less elegant PipeDream 3 commands don’t appear in the PipeDream 4 menus nor in the PipeDream 4 User Guide—but they still work. Let me give you an example.

The PipeDream 3 command <Ctrl ENT> (Edit Number to Text) is used to convert the numerical contents of a cell to text. It also converts formula to the textual equivalent of the formula. It does not convert the formula in the cell to the value returned by the formula. This command is absent from the PipeDream 4 menus and User Guide—but it works!

The <Ctrl ENT> of PipeDream 3 was replaced by <Ctrl BNT> (Block Number to Text) in PipeDream 4. The logic behind this change is that the command, even in PipeDream 3, operates on marked blocks as well as single cells. The use of “Edit” rather than “Block” was considered to be misleading or at least, unhelpful. Nevertheless, even though it does not appear in the menus nor in the User Guide, if you execute <Ctrl ENT> in PipeDream 4 this undocumented command still works exactly as it did in PipeDream 3.



PipeDream File



Command File

I do not intend to list the undocumented commands which still exist in PipeDream 4; there are many. They are retained as an undocumented features of PipeDream 4 so that Macros created using PipeDream 3 commands still function as Command Files in PipeDream 4.

What is a Command File?

At last we can get started! A Command File is a series of commands. One way of looking at it is as a program built from PipeDream commands. It can be executed in any of three ways: either by the “Do Command File” command, <CTRL FZ>; double clicking on the Command File; or by dragging the Command File into an open PipeDream document.

What are Command Files Used for?

If want to enter the same text many times; or if keys, including function keys, are to be redefined; or if you often require the same sequence commands; then you should create and use a Command File.

Command Files can be used to change the shape and content of one or many PipeDream documents. For example you can change the width of columns, add or delete columns and rows, mark and copy blocks (even from one file to another), you can even load and save files and you can do much more. Anything which you can do directly from the keyboard using PipeDream commands can be automated into a Command File.

It is almost impossible to this with Fireworkz since Command Files are not sufficiently well documented and certainly they are not as comprehensive as they are in PipeDream.

Creating Command Files

Although it is possible to construct a Command File by direct entry of commands into a text editor such as !Edit or even !PipeDream, it is much easier to use the Record Command command, <Ctrl FY>. Some commands are lengthy and their syntax is not intuitive. Mistakes are easily made. Even if I am working on a Command File in !Edit or in !PipeDream, I find the simplest way of ‘discovering’ the syntax of the command is to record it and then Copy and Paste from my recording into the file I am constructing before continuing to edit that Command File.

A recorded Command File will be stored in the ‘current PipeDream directory’, that is in the same directory as the currently open PipeDream file. The full Path to the currently open PipeDream file (the top of the window is yellow) is extracted by PipeDream and used to point to the directory in which the recorded Command File is stored.

When you execute <Ctrl FY> a dialogue box will pop up and you will be asked to insert a file name for the Command File. PipeDream will create a file with FileType Data and it will appear as an empty file in the directory window.

In many simple cases all you need to do now is to execute carefully all the commands you wish to record. Of course there is a knack to it. With practice you’ll find better ways of doing what you want to do in such a way that the recorded Command File will work on a whole range of different files requiring the same treatment. When you have finished recording you close the Command File with the same command that you used to open it, <Ctrl FY>. The file icon will change from the Data icon to that used for PDCmdFil files.

The next time you wish to execute that set of commands then, instead of repeating the process manually, you can use the Command File in any of the three ways I've described.

Don't Click on a Cell

A common mistake when recording is to use the mouse to control the position of the input focus. The PipeDream recording process does not recognise such actions. You must use commands such as <Ctrl CGS> (Caret Goto Slot) or you may use the arrow or <Tab> keys.

Generally <Ctrl CGS> is more economical than painfully moving the input focus down one row at a time using the cursor keys. Every movement of the input focus is a command and each movement will be faithfully recorded so, if you use the cursor keys instead of <Ctrl CGS>, then you will have many commands instead of one more efficient command. Perhaps you begin to see what I mean when I suggest that there is a "knack to it".

Full Path Names

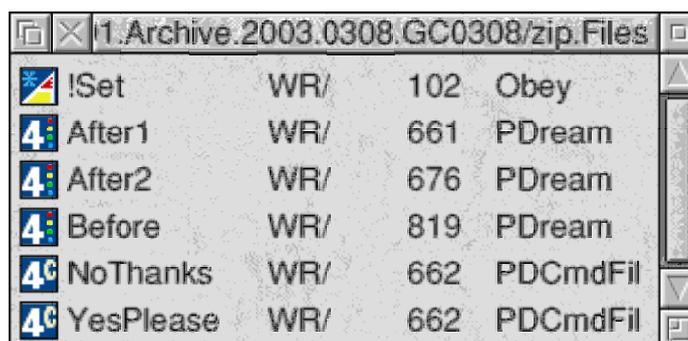
Another difficulty which arises when recording Command Files occurs because the recorder records the full path name to Loaded or Saved files. In many cases this is OK but in others it leads to an error. I have created the example below which Loads and Saves files. The path name I have used (which appears in the screenshot) will not be the same as the path you use if you try out this same example from the Archive monthly disc. The use of an inappropriate full path name will cause the Command File to return an error.

The System Variable <PipeLine\$Dir>

I have introduced a System Variable called <PipeLine\$Dir> through an Obey file called [!Set] (on the Archive monthly disc) which guarantees that the Command File of my example will run from anywhere on any disc on any RISC OS machine.

Availability

The files I am using for this example are not hard to create. However, they are available in many places including the Archive monthly disc and the Archive and Abacus Training Web sites. I suggest that you download them and run them rather than construct them.



Icon	Filename	Permissions	Size	Type
!Set	!Set	WR/	102	Obey
After1	After1	WR/	661	PDream
After2	After2	WR/	676	PDream
Before	Before	WR/	819	PDream
NoThanks	NoThanks	WR/	662	PDCmdFil
YesPlease	YesPlease	WR/	662	PDCmdFil

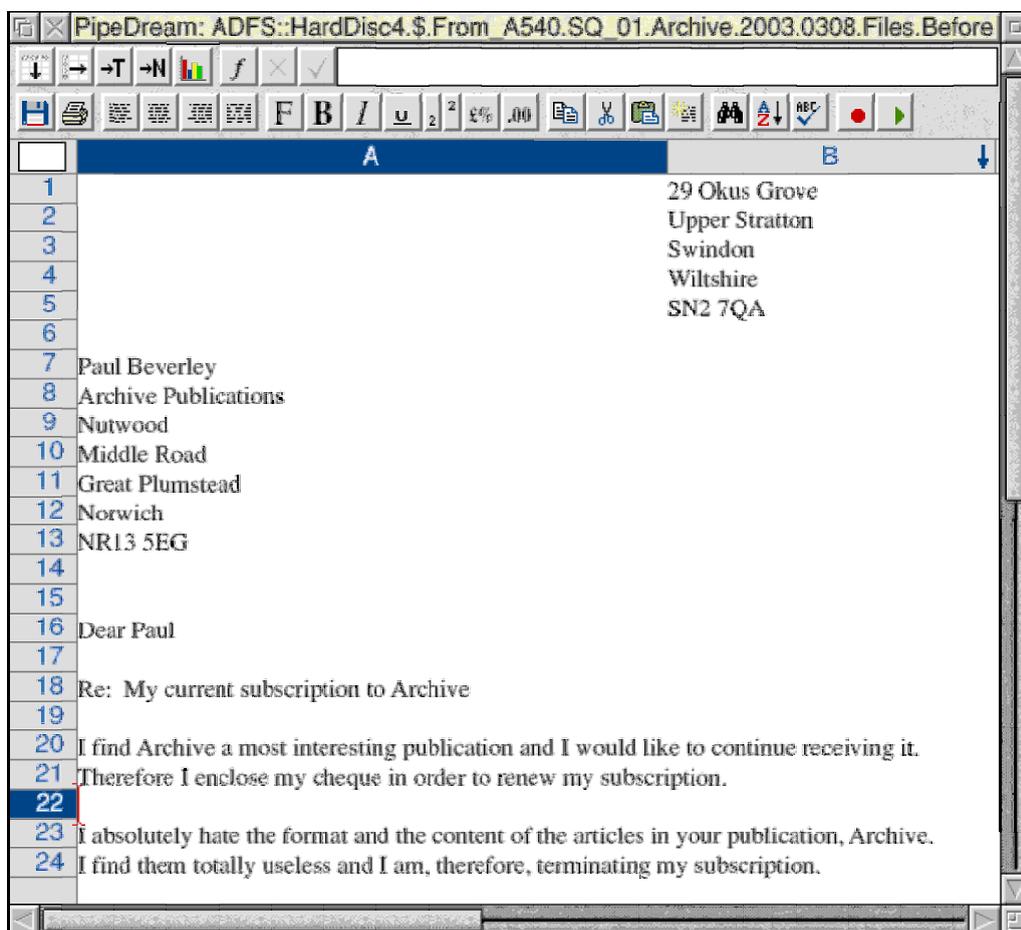
The Obey File [!Set]

There is only one important line in this Obey file. It is:

```
Set PipeLine$Dir <Obey$Dir>
```

You must double click on the Obey file [!Set] in order to create the System Variable <PipeLine\$Dir> before you try out the example. This System Variable is the full path to the same directory as that containing [!Set]. The System Variable <PipeLine\$Dir> appears in the PipeDream Command Files in front of the leaf name of the Loaded and Saved files.

The File [Before]



I have to thank Albert Kitchenside for the inspiration behind this example. Nevertheless, in recording my thanks to him I must make it clear that neither he nor I can be responsible for any inappropriate use of the content therein! In other words "Don't quote me!"

I accept that it is a somewhat artificial example. I use something similar but much more complex to insert customer addresses and appropriate clauses into a standard letter.

There are two things I want you to notice about this 'form letter'. The first is that there is no date in cell B7 and the second is that the content of the two rows 20 & 21 is an alternative to the two rows 23 & 24.

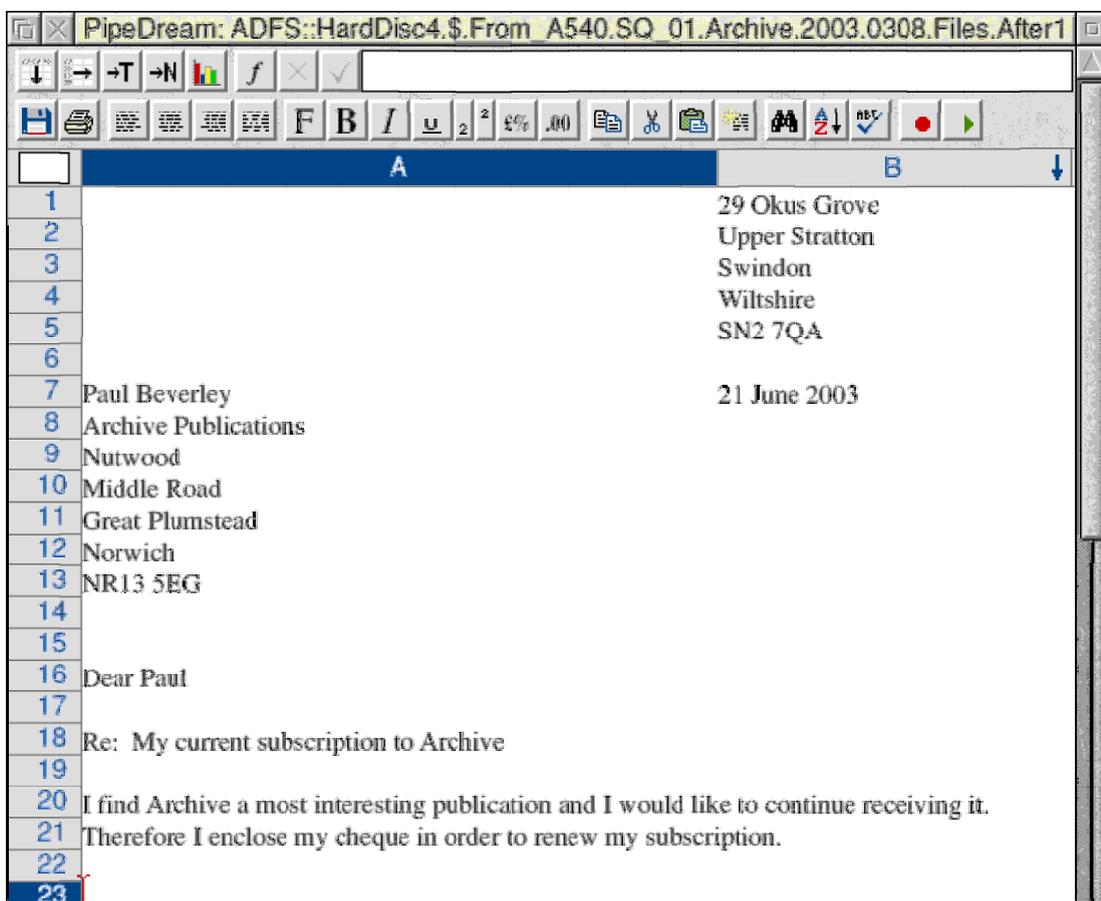
The File [YesPlease]

I have included two example Command Files for inclusion with this form letter. As you might expect, the file [YesPlease] renews my subscription whereas [NoThanks] terminates my subscription. Both Command Files enter today's date in cell B7.

The command file [YesPlease] is in ASCII text format but it has the FileType PDCmdFil so that double clicking on it will cause PipeDream and not !Edit to execute the file.

```
\Fl |i "<PipeLine$Dir>.Before" |m Load File
\Cgs|i "B7" |m Go to slot
@D@
\Z|m Mark block
\Bss|m Snapshot
\Q|m Clear markers
\Cgs|i "A23" |m Go to slot
\Z|m Mark block
\Ccd|m Cursor down
\Z|m Mark block
\Bd|m Delete block
\Y|m Delete row
\Fs|i "<PipeLine$Dir>.After1" |i N |i "" |i N |i "LF" |i P |m
```

This [YesPlease] Command File Loads the file [Before], enters today's date in B7, deletes the 'termination' clause and Saves the file as [After1], shown in the screenshot.



The [NoThanks] Command File Loads the file [Before], enters the date in B7, deletes the 'renewal' clause and Saves the file as [After2].

Testing

If you have the files from one of the the websites or the monthly disc then you can run this example. First double click on [!Set] to create the system variable, then double click on either [YesPlease] or [NoThanks] and that's it! Check the new files [After1] & [After2].

Expansion

I am sure that you can visualise how this technique can be extended to Load any address (taken from a marked block in a data file) to A7A13 using the 'Insert at slot' parameter of <Ctrl FL>. It is a little more difficult to patch together standard clauses selected by using a set of key words, but, with a little practice and patience it can be done.

Conclusion

Command Files are different from Custom Functions. Custom functions execute a sequence of PipeDream functions whereas Command Files execute a sequence of PipeDream commands. Custom Functions can not be used to change the shape of a document whereas Command Files can do this and can even merge documents.

The Command File record command, <Ctrl FY>, takes all the hard work out of the creation of Command Files. The syntax of every command is always perfect!

There are pitfalls to look out for such as avoiding mouse movements and being wary of the consequences of recording full path names to files that you may want to move around your disc. With practice these pitfalls can be avoided or worked around. PipeDream 3 Macros and PipeDream 4 Command Files are worth the 'learning curve'.

Contact

You can contact me by email at Gerald L Fitton <GoldLine@abacusline.demon.co.uk>. Files can be downloaded from <http://www.abacusline.demon.co.uk/Archive/GC0308.zip> and are available on the Archive monthly disc.