Gerald's Column
by Gerald Fitton

Last month my subject was 'Filters'. Filters allow you to convert files created in one package (such as PipeDream) into the format of another package (such as Impression). With a PipeDream to Impression filter you can create a table using PipeDream, convert the file to Impression format and Load it into Impression. As managers of Colton Software products we shall encourage third parties (such as you) to write filters for PipeDream and Fireworkz (for example a PipeDream to HTML filter). I believe that filters for converting the formats of other popular packages (such as Schema, Impression, Ovation, Excel and Word) will be developed by third parties rather than the owners of those packages.

There is another process called Object Linking and Embedding (OLE) which allows data from one package to be displayed in and printed from another package. In my articles this month and next month I shall describe what OLE is and how it works. If you haven't used OLE then I suggest that you experiment with my examples. Like most techniques, once you know how to use OLE, you'll find it easy to use. It is a most rewarding technique to learn because you will be able to mix your use of word processor, spreadsheet, graphics and Desk Top Publisher packages to produce better results more efficiently than before!

I was always under the impression that OLE was the invention of someone at Acorn but, during my research for this article, several of my correspondents have told me that it was Computer Concepts who first introduced this useful concept.

Disc Files

I do recommend that you experiment with the examples I have provided on disc. All the files used in these two articles appear on the Archive monthly disc, on the Archive web site and on our own web site. Eventually they will appear on the Archive annual CD. If you want these examples and are unable to get them by any other means then please write to me enclosing a preformatted floppy disc and return postage; I will send you the files.

History

It is always useful to study history. Looking at the way in which a technique has been developed encourages lateral thinking and illuminates the pathway to further progress.

I have found it difficult to trace the origins of OLE. I've discovered a useful article buried within the Impression package which (if you have Impression) can be accessed through the menu by selecting Help – Index – OLE. Also George Buchanan, the author of TableMate, (which supports OLE) gave me a web site address containing a document authored by a certain 'Mike' (surname unknown). Mike started to write a "Specification of generic 'OLE' interface and mechanisms" in August 1993. I can not trace any indication that this specification (see monthly disc) has been modified since the original was penned in 1993.

Although my historical research is incomplete it is probably fair to credit either the somewhat anonymous 'Mike' or some other employee of Computer Concepts with the invention of the OLE concept. Please let me know if I am wrong. Also, please let me know if you have more recent information about the OLE specification.

What is OLE?

The official statement (by Computer Concepts) is that "Object Linking and Embedding (OLE) is a technique which allows an 'object' from one package to be embedded in another". However, as we shall see, it has become more than that.

The first of the two packages is used to create and edit what Computer Concepts (and Acorn) call 'objects'; a simple example of an 'object' is a spreadsheet table created using TableMate. The second package is used to display this 'object'; this second package is usually a Desk Top Publisher such as Impression. In this example the TableMate spreadsheet table would appear in and become a part of the Impression document.

For the purpose of these articles I shall refer to the first package as an Editor and I shall refer to the second as a Desk Top Publisher (DTP). In Mike's specification (included on the Archive disc and elsewhere) he refers to the Editor as the "Server" and the DTP as the "Client". The "Client" application does not have to be a Desk Top Publisher.

Once you have embedded the 'object' in the DTP file you can send the DTP file to a friend who does not have the Editor application. Your friend will be able to Load the DTP file, view the embedded object and print the whole DTP file including the embedded object.

Using my example your friend will be able to print your Impression format file. The TableMate table you have created will be printed as part of the Impression document.

With OLE you can do more than this. If your friend has the Editor application then they can recover the object and edit it using their own copy of the Editor. In my example if your friend has TableMate they will be able to extract the TableMate file, Load it into their own copy of TableMate, edit the file and return the amended table to Impression.

Draw Files

Before we go any further I must make the distinction between Saving as a DrawFile and Saving in what I shall call Draw+Data format.

I have used for my examples of Editors the three spreadsheets Eureka, TableMate and TableCalc but in doing this I do not wish to imply that the use of OLE is limited to linking Spreadsheets with Desk Top Publishers.

Example files in these three formats are on the Archive monthly disc and elsewhere.

All the spreadsheet packages I know of allow you to Save in foreign formats. The three I have named allow you to 'Save as DrawFile'. These DrawFiles can be loaded into any package which can display DrawFiles (such as PipeDream and Ovation) and they can be edited using any package (such as Draw, DrawPlus and Vector) which can edit DrawFiles.

I have Saved the graphic shown below from TableCalc as a DrawFile. In this article (as I have sent it to our editor) I have Loaded the DrawFile into an embedded Impression frame.

		Item	Category	Sub Category	Cost
		1	Food	Meat	£3.50
		2	Clothes	Shoes	£31.32
		3	House	Gas	£43.00
		4	Clothes	Socks	£4.50
		5	Food	Veg	£2.46
		6	House	Electric	£34.67
		7	Clothes	Suit	£65.37
		8	House	Water	£23.68
		9	House	Rates	£25.38
		10	Foood	Salad	£4.72
		11	Clothes	Hat	£19.99
				Total	£258.59

By the way, before you pick up your pens (or hit your keyboard) I do know that "Food" is not spelt as "Foood" (see item 10). The misspelling is part of my example.

If the Impression file which I sent to Paul were your Impression file and if you were to send it to your friend (the one without TableCalc) then they could recover the DrawFile, Load it into their copy of Draw and change "Foood" to "Food" before printing the file. Paul, our Editor, could do this too—but I hope that he won't!

One of the limitations of the current version of Fireworkz (V 1.31/05) is that it does not support 'Save as DrawFile'. If you or I wish to include a Fireworkz table in a DTP document, then we have to use a screenshot (or a filter—see last month's article). Screenshots are sprites. Sprites are large files with limited resolution whereas DrawFiles are much smaller and have almost unlimited resolution. You can select part of a DrawFile and include just your selection in the DTP and you can do so at any scale you choose.

The good news for those of you with Fireworkz is that it will have 'Save as DrawFile' added in the near future (check our website if you want to monitor progress).

Draw+Data Format

For an Editor to have an OLE facility it must be able to Save in Draw+Data format. Eureka, TableCalc and TableMate files can be Saved in Draw+Data format.

On the Archive monthly disc I have included files in Draw+Data format which I have Saved from Eureka, TableCalc and TableMate. I suggest that you experiment with these Draw+Data files. You do not need any of these Editor packages to do some of the experiments which I suggest (some experiments this month and some next month).

The Draw+Data format consists of two sections.

The first part is a DrawFile representation of the Editor file. If the Editor package can already create DrawFiles then this first part will be the DrawFile which would be Saved when the 'Save as DrawFile' option is selected.

The second part of the file contains data in the format used by the Editor. For example the second part of a Eureka Draw+Data file is in standard Eureka format. This second part is contained within something called 'draw tags'.

Display

When a Draw+Data file is Loaded into a DTP it will be embedded within a frame. The DTP will display the DrawFile which is the first part of the Draw+Data file and it will preserve (but not use) the second part of the file. You will remember that this second part of the Draw+Data file contains data in the format used by the Editor package.

Draw+Data files can be loaded into (almost) any package which accepts graphics in DrawFile format. The first part of the Draw+Data format file is in DrawFile format and it is this DrawFile which will be displayed.

Fireworkz does not support OLE but the Draw+Data format files will Load into Fireworkz and will be displayed as graphics—they are treated as DrawFiles. On the Archive monthly disc I have included a file in Fireworkz format called [!Fire1] which contains and displays three Draw+Data files created using Eureka, TableCalc and TableMate.

The graphic shown below is a screenshot of the TableMate table taken from the Fireworkz file. It is a sprite with a file size about ten times larger than that of the DrawFile graphic of the TableCalc file shown earlier in this article. Note the poorer resolution.

	Item	Category	Sub Category	Cost
	1	Food	Meat	£3.50
	2	Clothes	Shoes	£31.32
	3	House	Gas	£43.00
	4	Clothes	Socks	£4.50
	5	Food	Veg	£2.46
	6	House	Electric	£34.67
	7	Clothes	Suit	£65.37
	8	House	Water	£23.68
	9	House	Rates	£25.38
	10	Foood	Salad	£4.72
	11	Clothes	Hat	£19.99
			Total	£258.59

Non OLE Recovery

The TableMate Draw+Data file is embedded in the Fireworkz file [!Fire1] and it can be recovered from Fireworkz by using the Extra – Pictures – Save picture... sub menu. On the monthly disc you will find this file; it is called [PicTM]. On the disc this file has the filetype DrawFile and it will Load into Draw and DrawPlus. However, if you change the filetype to TblMate then you'll find that you can Load it into TableMate!

I shall return to the subtleties of what formats can be Loaded into which packages next month. For now what I want you to recognise is that the file [PicTM] Saved from Fireworkz (a non OLE package) will Load into both Draw and into TableMate. It is the DrawFile part of the Draw+Data format file which is displayed in Draw and the Data part of the file which is used by TableMate. If you Load the [PicTM] file into !Edit (hold down <Shift> as you double click on the file) you will see that the file starts with:

```
Drawé[00][00][00][00][00][00][00]TableMate
```

You can use this 'Load into !Edit' technique to check the original Editor package used to create a Draw+Data file. They all start with `Draw` and they all include the originating package in the position you see the word `TableMate`.

OLE Recovery

On the Archive monthly disc (and elsewhere) you will find files in Impression and Ovation format which contain the three Draw+Data files created using Eureka, TableCalc and TableMate. Both Impression and Ovation support the OLE protocol. If you have either Impression or Ovation and if you have one of the three Editor packages then you will be able recover the Draw+Data file from the DTP file and edit it.

The convention for extracting the Draw+Data file is with <Ctrl double click> on the DTP graphic frame containing the Draw+Data file. By <Ctrl double click> I mean hold down the <Ctrl> key and double click the mouse <Select> (left) button on the graphic frame.

If you do this on, say, the frame containing the Eureka table then Eureka will Load to the icon bar (unless it is already there) and the Draw+Data file will appear in a Eureka window ready for editing. You will be able to change "Foood" to "Food" (and much more) before returning the amended file to the DTP using the usual Save process of Eureka (for example tap <F3> followed by <Return>).

Summary

I have deliberately avoided equating the Draw+Data format with OLE. Fireworkz does not support the OLE protocol (which uses <Ctrl double click> for recovery of the 'object') but does allow Draw+Data files to be Loaded and displayed in it as well as recovered from it.

At the heart of my explanation and description of OLE is an understanding of the format I have called Draw+Data. Draw+Data format files contain both a graphic in DrawFile format and data in the format used by the Editor application.

The DrawFile part of the Draw+Data format file can be displayed in (almost) any package which will display DrawFiles. The Draw+Data 'object' can be stored in (almost) any package which can 'embed' DrawFiles. By 'embed' I mean that the Draw+Data file is Saved as part of the file in which it is 'embedded'. The Draw+Data 'object' can be recovered from (almost) any package in which it can be 'embedded'.

The OLE protocol is something extra. If a package supports the OLE protocol then Draw+Data files are recovered by using the <Ctrl double click> technique..

Next Month

Up to now I have described features which are (almost) identical for all Editor and DTP packages which support the OLE protocol. You will not be surprised to learn that different Editors and different DTPs behave in slightly different ways during Saving, Loading, Recovery and Editing. I shall describe these differences next month.

In the meantime, if you do experiment with the files and want to make any comments about this technique or add to my knowledge of the subject then please write to me or email me. Please don't telephone! By the way, my fax machine is now permanently off-line because I am receiving too many junk faxes from anonymous sources.

Finally

Thanks for your many emails and letters. Thanks also to those who have sent me a self addressed sticky label and return postage with their letters. Again my thanks to those who have included an example illustrating their comments or difficulties as an email attachment or on floppy disc. My address is that of Abacus Training which you will find in the Fact File at the back of Archive.