Gerald’s Column
by Gerald Fitton

A long time ago the <End> key was marked <Copy> and it could be used to copy text from the ‘pointer’ position to the input focus. Later this key was marked <End> and <Copy> was dropped. For many of the earliest Acorn packages this <End> key was defined as ‘move the input focus to the end of the file’. Later packages, notably Impression, defined its role as ‘delete the character to the right of the cursor’.

If you use a Windows machine then you will know that the <End> key is used as an ‘end’ key and the <Delete> key is used to delete the character to the right (not to the left) of the cursor. The <Backspace> key is used to delete the character to the left of the cursor. Currently the ‘in word’ is “convergence”. In the context of keyboard shortcuts the fashion is to change current definitions so that they are more familiar to a Windows user. This means <Delete> deletes to the right and not to the left; <End> means ‘end of file (or row)’.

Last Month

Last month I explained how you can use PipeDream’s [Key] file to keep up with the latest fashion in keyboard shortcuts. For example the <Delete> key can be defined in such a way that it deletes to the right rather than the ‘old’ RISC OS standard which was to delete the character to the left. By default PipeDream uses the <End> key as an ‘end’ key; this is in accordance with the latest fashion in ‘convergence’. Nevertheless, if you want to change it to become a ‘delete right’ shortcut then the [Key] file can be used to do this trick.

This Month

There is a little known and little used facility which is built into Fireworkz called a User Library. A User Library can be used to store regularly used Graphics files and Custom Functions; these can be loaded into any document. The Library facility exists in PipeDream 3 but not PipeDream 4. This month I shall explain how you can add such a Library facility to later versions of PipeDream.

Location

Following the convention used by Fireworkz, the most appropriate place for a User Library in later versions of PipeDream is within the User sub directory so that it has the full path name !PipeDream.User.Library. In the earlier versions of PipeDream such as PipeDream 3 the Library subdirectory has the full path name !PipeDream.Library.

On the Archive monthly disc I have included a dummy !PipeDream application which contains a User sub directory. In this User sub directory I have included a graphics file called [Xmas] and a couple of custom functions as my examples.

If you copy the file [Library.Xmas] to your copy of !PipeDream (using the directory structure which I have) then you will create a !PipeDream.User.Library containing these example files.

The File [Xmas]

The file [Xmas] contains a colour graphic. It is in the form of a DrawFile containing the three pictures shown in the graphic below.

[Xmas] is intended as a simple example of files which you can put into your Library. In my Library I have many graphic files such as my letterhead and some regularly used custom functions such as [c_Stats].

Referencing [Xmas]

Also on the monthly disc you will find a file called [TestFile1]. This PipeDream file contains only one entry in cell A3. Cell A3 contains @G:Xmas,100@.

Those of you familiar with PipeDream will recognise this as an instruction to PipeDream to include the graphic file [Xmas] as a ‘Referenced Document’ and to include this graphic file at 100% scale. If your use of [Xmas] is a ‘one-off’ then the most appropriate place for [Xmas] is in the same directory as the file [TestFile1]. In my User Library I have such graphic files as my letterhead. I do not want to include this letterhead graphic in every directory containing a letter; I don’t need to because it is in my User Library.

What is the Search Path?

When a PipeDream file contains a referenced document (such as [Xmas]) the first thing it checks is whether the document name is a full path name or only a ‘leaf name’. If the name is a full path name then that is where PipeDream looks.

However, if, as in our case, the leaf name, [Xmas] is all that is present then PipeDream will look in the same directory as the file from which the referenced document is called, in our case this is the directory containing the file [TestFile1]. [Xmas] isn’t there!

Then PipeDream looks in the User Directory.

In this Directory it will find your personal [Choices] and your User Dictionary. There are many other User Directories such as your Templates, your Markers (the shape and style of points for graphs), Pictures (which are used by Charts for Pictographs) and the newly added Library.

We could have placed the file [Xmas] in the User directory but, if, like me, you have a lot of regularly used files then this would soon ‘clutter up’ the User directory. It is better to use the technique built into Fireworkz—a User Library.

Finally PipeDream looks in the Resource Directory.

As a general rule I suggest that you don’t ‘mess around’ with anything in the Resource Directory. Nearly all the files in the Resource directory are necessary to the proper operation of PipeDream and it is a bad idea to wreck anything in there. You can’t do much damage if you limit your activities to the User directory!

The User Library

In order to persuade PipeDream to look in the User Library an extra instruction must be added to the [!Boot] file. Within the skeleton [!PipeDream] application on the Archive monthly disc (in the [Files] sub directory) you will find a version of the [!Boot] file which includes such an instruction. The important (modified) line is:

```
 SetMacro PipeDream$Path
             <PipeDream$Dir>.User.,
             <PipeDream$Dir>.User.Library.,
             <PipeDream$Dir>.Resource.
```

Be careful will all the full stops and commas. They are all important! Also note that there is a space between PipeDream$Path and <PipeDream$Dir>.User,. What I have done is to include an extra directory, User.Library (the third line), in this path. I couldn’t get the whole instruction on one line so I have split it into four lines. Don’t do this in the real [!Boot] file. Look at what you already have and add in the User.Library as I have done.

The system variable PipeDream$Path is evaluated every time PipeDream looks for a file (a Referenced or Dependent Document—and a few other things as well).

You will see that files in the User Directory take precedence over those in the Resource directory. This is the correct order of precedence. The Resource directory contains default versions of, for example, templates, which you might wish to change. The revised template will be in the User directory but the original will remain in the Resource directory should you wish to revert to it.

What I have done is to insert an instruction to look in the User Library before looking in the Resource directory. If you replace your current [!Boot] file with this one, or if you modify your [!Boot] file by adding a ‘call’ to a User Library, then files in the Library such as [Xmas] will be found.

Custom Functions

Your newly formed User Library can contain not only referenced (graphic) files such as [Xmas], but also dependent documents such as custom functions or regularly used data files (and a few other interesting types of file).

If you build up your own Library of custom functions then PipeDream will find them when you need them without you having to load them separately or specially.

On the Archive monthly disc, in the User Library I have included a demonstration custom function called [c_Add]. This custom function adds 1 to the number passed as a parameter to the custom function. On the Archive monthly disc I have included [TestFile2] which contains the formula [c_Add]add_anything(5) so that the number returned is 5 + 1 = 6.

Note that the file containing the custom function, [c_Add], is a dependent document and not a referenced document as is [Xmas]. PipeDream files this dependent document just as it found the referenced graphic document [Xmas].

As another example I have included the [c_PostCode] custom function which will allow you to print a post code in bar code format.

A long time ago Stephen T Brown wrote a set of statistical custom functions which, over the years, I have added to myself. You will find these in [c_Stats]. If you copy these into your User.Library then you will be able to call them from any PipeDream document.

Summary

Both Fireworkz and PipeDream 3 contains a Library facility which is not present in later versions of PipeDream. However, by modifying the !Boot file, PipeDream can be instructed to look in a User Library for commonly used files in the same way as PipeDream 3 and Fireworkz.

The User Library can contain referenced and dependent documents. These can be graphic files such as a letterhead or they can be custom functions such as a compendium of statistical functions. The Library can also contain command files—but that’s another story.