ARCHIVE PROGRAMS **Dilwyn Jones**

Recently, I discovered a copy of an old DJC program I thought long lost, the Cocktails Waiter program. Its original author (Imre Dominik) had allowed me to release his programs as freeware as I saw fit, but its original program disk was corrupt and there was no source code disk surviving at all. The rescued copy was on an old set of backup disks I discovered as I was "recycling" some old floppy disks from the attic. Not all of the disk was readable, but I managed to get the program "as sold" and part of the manual, which I rewrote from scratch which was easier than fiddling with the corrupt Quill DOC file.

The main problem was the lack of source code. Cocktails Waiter was written in Archive with a large controlling program and a 400-plus record database, supplied in a protected format using the Psion Runtime Archive system. The database turned out to be fine - that could be read with standard Archive. The main "BOOT_PRO" program unfortunately was protected and so could not be loaded into standard Archive for viewing, updating and saving, as it had been saved as protected object code for use with the Archive Runtime Module, a special version of Archive developed by Psion in the 1980s for software distribution. There followed a brief discussion on the mailing list about this and enough information was gleaned to salvage something from all this.

John Gilpin asked me if I'd write up some of this for the newsletter, as it might prove useful information, so here goes, starting with the various formats concerned.

Programs written in Archive can be saved to disk in any of three formats:

1. _prg is a plain text Archive program, saved from Archive with save'filename_prg'

2. _pro is an "object" (tokenised) program which can be protected or unprotected. _pro files can be saved in two ways:

save object'filename_pro' which gives an unprotected tokenised or binary program
save protect'filename_pro' which gives a protected binary program

The only difference between the two _pro types is that the byte at file position 6 is 0 for unprotected, 1 for protected.

Here's a short SuperBASIC program which will "unprotect" a protected Archive program:

```
100 REMark unprotect a protected Archive _pro file
110 CLS : CLS #0
120 INPUT #0,'Enter name of Archive _pro program to
unprotect > ';ip$
130 OPEN #3,ip$
140 BGET #3\6,byte
150 IF byte = 0 THEN
160 PRINT #0,'Already unprotected.'
170 CLOSE #3 : STOP
180 END IF
190 BPUT #3\6,0
200 CLOSE #3
210 PRINT #0,ip$;' now unprotected.'
```

The program simply resets the byte at file position 6 - this is the "protect" flag. It was a feature built into Archive allowing commercial programs to be protected. The program was saved in a tokenised form unreadable if you tried to view it on screen.

I think that the main advantages of _pro programs over _prg ones are:

1. slightly smaller, as they are tokenised.

2. slightly faster loading, as no need to tokenise at loading time,

and

3. the program cannot be viewed in the normal way by copying to the screen for example.

The advantages of _prg program files was that they were plain text, so easier to read outside of Archive in an editor if you wished (e.g. Quanta newsletter editors could import them direct into their wordprocessors!) and they worked on all versions of Archive, whereas the _pro files could vary slightly between Archive versions, especially between QL and PC versions, causing some loading difficulties if loaded into a different version of Archive than that in which they were created (as I found to my cost, until I found a copy of the right version of Archive).

The "Runtime Archive" was a special version of Archive released by Sinclair and Psion to allow software developers to do what Imre Dominik did, which was to write Archive-based programs with the option to protect their code from view. The Runtime Archive had a few extra facilities compared to standard Archive, e.g. machine code interface, line graphics characters, automatic use of full screen on a QL (no prompts windows, for example), more free memory, and facility to automatically run a boot_pro file if it had a procedure called start and so on. The issue was slightly clouded by the fact that some of these features made their way into later versions of Archive, but were not necessarily consistent across versions.

So, as Cocktails Waiter was supplied as a Runtime Archive suite with protected code, I had to take several steps to recover an editable program source:

1. Unprotect the boot_pro program with the above listing.

2. Find a version of Archive which would load the unprotected boot_pro. I have a few versions of QL Archive and Xchange - it was eventually Xchange Archive which I used with greatest success.

Luckily, Imre did not use much by way of code specific to a particular version of Archive, apart from using multi-line fields in the Archive screen which seem to be specific to the Runtime Archive system - it has its own version of

Archive called ARCHDEV for writing programs specifically for use to be distributed with the runtime system.

3. Check that all the code looked OK by means of viewing it in the Archive editor, using TAB to step through the various procedures, and listing it out to see if there was any garbled code in there.

4. Save it out as a _prg file using the command save "flp1_boot_prg" - this I have kept as the source code version, which I can amend should any changes prove necessary at some point.

When Psion gave permission for the QL versions of Archive, Quill, Abacus and Easel to be freely distributed, I don't know if that included the runtime archive development system. I have a copy of version 2.32 and its manual so could make it generally available if permitted. It is certainly interesting and the manual is useful as it documents the machine code interface (lets you add user written machine code routines to Archive, such as commands to add sound) and other nice little extras like the line graphics characters and control character screen printing available in later versions of Archive.