



## Data Access Corporation v Powerflex, Bennett & Ors

*Jim FitzSimons*


This was a decision of Jenkinson J, sitting in the Federal Court of Australia, Melbourne registry. Judgment was handed down on 9 February 1996.

The facts in this case were quite straightforward and appear not to have been in issue between the parties in most respects.

The Applicants owned the copyright in the application development system, DataFlex. Dr Bennett is a medical practitioner who became very familiar with DataFlex whilst working as a Fellow in Computer Medicine. Starting in 1985, he set out to create an application development system which would be highly compatible with DataFlex. Both

DataFlex and the program created by Dr Bennett are Fourth Generation Languages.


Dr Bennett never had access to the source code of DataFlex, nor did he seek to decompile it. His program was the product of a separate effort

*Continued on page 3* 

### ***In this issue...***

#### ***Computer Copyright Update***

Data Access Corporation v Powerflex, Bennett & Ors by <i>Jim FitzSimons</i>	1	Legal aspects of computer programs security in Spain by <i>Enrique Batalla</i>	28
Data Access Corporation v Powerflex Services Pty Ltd by <i>Ashely Porter</i>	8	Planning and implementing information systems in legal practice by <i>Thomas Bassett</i>	32
Data Access Corporation v Powerflex Services Pty Limited <i>Ron Eckstrom</i>	9	Copyright Act update & maximising copyright protection by <i>Philip Catania</i>	40
Licences of copyright works and the Internet by <i>Dayan Quinn</i>	11	Copyright protection of compilations of Acts by <i>Anne Trimmer</i>	44
Other ways of protecting computer software — what are the viable alternatives to copyright? by <i>Jane Owen</i>	13	Internet News	46
Copyright and databases by <i>Matthew James Mullan</i>	19	Abstracts	48
Indian giving?: Shareware and the Trumpet Software Case by <i>Kate Watts</i>	25	Case Note by <i>Stuart Marks</i>	49
		Press Release	50

 Continued from page 1

designed to produce a program which closely emulated the language and file structure of the original. He called it PFXplus and formed the company Powerflex to market and support it.

According to Dr Bennett, in a conversation with the writer, PFXplus does not emulate the look and feel of Powerflex; different colours are used, the manuals are different as is the packaging. A user would have no doubt about which program they were using.

Between 1989, when Powerflex released PFXplus, and 1993 when the present action commenced, the two programs coexisted in the market and various conversations between the principals of Data Access and those of Powerflex concerning the rival products took place.

DataFlex provides a language which, apart from the words associated with

graphics, comprises 225 words. Of those 225 words, 192 are in PFXplus. Use of these 192 words in either language invokes a procedure which causes a device having digital information processing capability (ie a computer) to perform the same function. However, there was generally no similarity between the source code of the two languages.

Having established these facts, His Honour analysed them in terms of the Australian *Copyright Act*.

### **The Words in PFXplus**

In its major finding, the court held that each of those words is a computer program under the Australian Act.

The Respondents referred to *Lotus v Borland* 49 F.3rd 807 (1st Cir. 1995) and argued that the function of the words in DataFlex is inseparable from the expression of those words and therefore that they are not entitled to copyright protection. The US

*Copyright Act* specifically states that a method of operation is not copyrightable (under S102(b)) and it was held in *Lotus v Borland* that specific words which are essential to operating something are part of a 'method of operation' and therefore are unprotectable.

Jenkinson J accepted that where the expression cannot be separated from the idea there would be no protection, but found that in this case the words used in DataFlex go beyond that and that each was an original expression and therefore subject to copyright. Each 'word', His Honour found, was a concatenation of letters, and the mere fact that the result resembled a single English word did not preclude it from copyright protection in its own right. He specifically stated that because the Australian Act contains no equivalent to s102(b) of the US *Copyright Act*, analysis concerning the 'method of operation' which is discussed in the American cases could play no part in his decision.



## COMPUTERS & LAW

### Editors

**Simon Pollard**

c/- Gilbert & Tobin  
50 Carrington Street  
Sydney 2000 Australia  
Tel: (02) 367 3103  
Fax: (02) 367 3111  
e-mail: [spollard@gtlaw.com.au](mailto:spollard@gtlaw.com.au)

**David Standen**

c/- Gilbert & Tobin  
50 Carrington Street  
Sydney 2000 Australia  
Tel: (02) 367 3059  
Fax: (02) 367 3111  
e-mail: [dstanden@gtlaw.com.au](mailto:dstanden@gtlaw.com.au)

**Kent Davey**

Australian Government Solicitor  
GPO Box 2853AA  
Melbourne VIC 3001  
Tel: (03) 9242 1273  
Fax: (03) 9242 1481  
e-mail: [kent.davey@ag.ausgovag.telememo.au](mailto:kent.davey@ag.ausgovag.telememo.au)

**Layout & design:** Jackie Fitchman

**Subscription:** \$32.00 per 4 issues

**Advertisements:** Inserts \$300.00; for advertisements within the journal, rates and information will be provided by the Editors on request. Articles, news items, books for review and other items of interest may be sent to the Editors. Journal contents may be reproduced if the source is acknowledged.

## **Error Text**

Both programs contained text which appeared on a user's screen at the appropriate time to inform the user of the type of error which has occurred. Although the error text in the two systems was very similar, His Honour found the error text in PFXplus not to constitute an infringement. His Honour found that the PFXplus error table did not reproduce a 'substantial part' of DataFlex because the DataFlex program could operate without the error table. The finding seems to be a corollary to the finding of Dawson J in *AutoDesk v Dyason* 173 CLR 330 that anything which is essential to a program is a substantial part of it.

His Honour also found that the error text in DataFlex was not protected as a literary work in its own right because the words used were one of a quite limited number of ways to express the idea concerned. The idea and its expression have become inseparable, or 'merged', and therefore 'uncopyrightable'.

## **Macros**

There are three macros in DataFlex, 'Report', 'Entergroup', 'Enter' which had as equivalents in PFXplus 'Report.PFA', 'Entgroup.PFA', 'Enter.PFA'. The DataFlex macros were encoded in an intermediate code called I-code whilst the PFXplus macros were in the PFXplus source code. In each case the macros compiled to make the computer perform the same complex function and a translation of the I-Code instructions to source code revealed an objective similarity between the two source codes.

His Honour found that the PFXplus source code of the three macros was an adaptation of the expression in I-Code of each of the three sets of instructions invoked by the three commands. This was despite the apparent independent derivation of the PFXplus macro source code.

## **Compression Table**

DataFlex employs, as part of its runtime program a compression table to facilitate the saving of space on storage media. To ensure compatibility PFXplus employed an identical table.

The judge seems to have accepted the Respondents' argument that because the compression table was not essential (the program would function without it) there was no reproduction of a substantial part of a computer program. However, His Honour ruled that the compression table was itself a literary work and protected in its own right. He dismissed the argument that making the table did not involve sufficient skill or judgment; the fact that the DataFlex table was produced by a computer program did not alter the fact that it is protectable.

Another argument put by the Respondents was that the only way to produce a program which is compatible with DataFlex was to have a program with an identical compression table and therefore the idea and the expression had merged. The court did not disagree that the concept of 'merger' was relevant, but ruled that there are many ways to compress data and that the mere desire of Dr Bennett for compatibility does not mean that there had been any merger.

This could be considered a serious blow for open systems. Presumably, without an identical compression table it would not be possible to swap compressed files produced by the two different systems.

## **Function Keys**

Apparently DataFlex uses 16 function keys and ascribes to each one a word suggestion of the function performed by pressing the key. However, the word is by no means the only one which could be used. Each of the sixteen words is reproduced by PFXplus and is allocated the same function.

The court ruled that each of the words was itself a computer program under the Australian Act because pressing the correlative key caused the computer to perform a particular function. The copyright in the words in DataFlex was infringed by the words in PFXplus.

## **File Structures**

Although the source code versions of the modules which produce the file structures are not similar, it appears from reading the judgment that the file structures produced by the two programs are identical. Dr Bennett, however, stated in a conversation with the author that the file structures are not identical but are sufficiently similar to be interoperable; apparently a skilled user can tell from looking at a file which program it was produced by.

The court rejected a submission that PFXplus reproduced the expression of a set of instructions in DataFlex because the two modules have the same function. But the court accepted that they represented an adaptation of the set of instructions which determine file structure.

His Honour's reasoning on this point was not entirely clear but seems to have been based on the view that PFXplus was a 'version' of the relevant part of DataFlex and was also a substantial part of DataFlex, because it involved nearly all of the relevant part of DataFlex.

## **Defences**

Various defences argued by the Respondents were dismissed. The reasoning included:

- The fact that certain parts of the programs may be denied protection in the US because of a failure to include a copyright notice can have no effect on the status of those parts under Australian law.
- The failure by the Applicant to take action against the Respondents for years after finding out about the program

did not give rise to an implied licence from the Applicant to the Respondents.

- There was no evidence to support an allegation that the Applicant had encouraged the Respondents to believe that no copyright existed in DataFlex.
- The failure of the Applicant to take action did not represent inexcusable delay on its part; the Applicant was aware of the costs of litigation and the uncertainties in any proceedings and so was entitled to wait until circumstances dictated the necessity of taking action. In this case those circumstances were the threat to the DataFlex market which PFXplus began to represent after several years.

### **Essential features**

This case is consistent with the finding of the High Court in *AutoDesk v Dyason* and takes Australian law in this area further from that of the United States. Perhaps this is nowhere more evident than in the treatment of the 'essential' features of a computer program.

In *Lotus v Borland* the Court of Appeal states:

*"If specific words are essential to operating something, then they are part of a 'method of operation' and as such, are unprotectable [pursuant to s102(b) of the US Act]."*

In this case, on the other hand, Jenkinson J stated:

*"But the reproduction of that part of the DataFlex program was not reproduction in relation of a substantial part of the program. The program could function without an error text table ..."*

which seems to imply that 'substantiality' and being 'essential' are the same thing, or that it is not protectable unless it is essential. This echoes the statement of Dawson J in *AutoDesk v Dyason* that:

*"... Widget C is a computer program and a substantial, indeed essential, part of that program is the look-up*

*table by reference to which Widget C processes the information which it receives ..."* [emphasis added]

Nowhere did Dawson J feel that it was necessary to justify any further his assertion that the look-up table is indeed a 'substantial part'.

In this case, Jenkinson J seems to have taken the matter somewhat further than Dawson J by implying that something cannot be substantial unless it is essential. Taken to its logical conclusion this finding has far reaching implications. It could not, for example be applied to a book.

It will be interesting to see if an appeal is lodged by the Respondents and if they do whether they pick up on the fact that on the second occasion that the High Court examined *AutoDesk v Dyason* the Chief Justice (Mason CJ) moved away from a link between being a 'substantial part' and being 'essential' being made.

Whilst the two analyses of a portion of code being 'essential' are not directly contradictory, they certainly head in opposite directions. The result is *Lotus v Borland* was the opposite to the result in this case; Borland's look alike spreadsheet was held not to infringe Lotus' copyright. On the other hand, whilst Borland spreadsheets may not be completely interchangeable with Lotus spreadsheets, the fact that spreadsheets created by one of the programs can be converted and read by the other leads to a conclusion that the difference in level of compatibility is only one of degree.

Australia also seems to be heading in a different direction from the UK and Canada. When you consider that in *Richardson v Flanders* [1993] FSR 497 and in *Carolian v Triolet* (O'Leary J, Ontario Court of Justice, 12 February 1993) developers who sold the copyright in a program and then later redeveloped very similar programs were by and large held not to infringe the copyright in the earlier work, life seems to be very much more difficult for a reverse engineer in Australia than anywhere else in the world. (The UK position can be said to have

altered somewhat since the finding in *IBCOS v Barclays* where a programmer who 'rewrote' a program after assigning copyright in the initial version was found to have infringed the copyright in the initial version, but the evidence of actual copying in that case was so overwhelming the result was bound to be different from that in *Richardson v Flanders*.)

### **Function as a determinant of breach**

A troubling aspect of the decision is the seeming re-emergence of the theory that a computer program can infringe copyright in another one if it fulfils the same function. In the first instance judgment of Northrop J in *AutoDesk v Dyason* his Honour based a finding of infringement squarely on a finding that Dyason's program fulfilled the same function as that of AutoDesk. This simple equation was expressly overturned in the Federal Court Full Bench judgment. The High Court explicitly did not follow the first instance reasoning in reinstating a finding for the Applicant.

In the present decision his Honour referred in four different areas (the DataFlex words themselves, Macros, File Structures and the Function Keys) to the fact that PFXplus fulfils the same function as DataFlex. Although in the case of the macros and the File Structures his Honour found that Dr Bennett had made an adaptation rather than a reproduction, there was nonetheless a finding of infringement which was based on the identity of function, given that there was no objective similarity between the source code of the two programs, other than in the case of the macros where it appears that the PFXplus source code was independently derived.

Any return to a test of similarity of function is a step in the wrong direction. If similarity of function is the test, it can be said that the second spreadsheet program infringes the copyright in the first one.

## **Copyright in a single word**

Perhaps the most surprising finding is that there is copyright in a single word. Even in *AutoDesk v Dyason* the relevant copyright work was 128 bits long - or about three English words.

There are numerous decisions in British copyright systems finding single words and short phrases are too insubstantial to enjoy copyright. In *Sinanide v La Maison Kosmeo* (1928) 139 LT 365 (CA) the words 'youthful appearances are social necessities, not luxuries' was found to be insufficient for copyright to subsist. In *Kirk v J & R Fleming Ltd* [1924] Macg. Cop. Cas. 44 an advertisement consisting of four commonplace sentences, was also insufficient.

As well, single invented words, even a word such as "EXXON" which in its creation expended enormous creative effort and research, cannot enjoy copyright; *Exxon Corp. v Exxon Insurance Ltd.* [1982] 1 Ch. 133. On the other hand, lists of meaningless words and codes have been found to be sufficiently substantial; see *DP Anderson & Co Ltd v The Lieber Code Co* [1917] 2 KB 469; *Ager v Collingridge* (1886) 2 TLR 291.

## **Definition of a 'computer program'**

This case further extends the definition of 'computer program' under the *Copyright Act*. The Australian courts have adopted a consistently wide interpretation of the definition. In *AutoDesk* a combination of a shift register and an exclusive 'or' gate was held to be a computer program. In this case a single word has been found to fit the definition of 'computer program' in the Act. The High Court in *AutoDesk* also had no difficulty in finding that comments included within a computer program form part of that program.

The finding whilst unusual is welcome insofar as it prevents the need for any definitional arguments when computer programs are the subject of actions before the courts.

On the other hand, the result in this

case might have been quite different in a very important respect if the court had found that the only relevant computer program was all the code as a whole. 192 of 225 words being the same certainly seems like a substantial portion. However, looked at against the program as a whole those 192 words were probably less than 1%

While the test of substantiality is one of quality not of quantity, there is no doubt that sheer quantity can be influential in a finding. This was seen in the first instance judgement in *AS2000 v CCH* where it was found that the extent of the copying was so great that it took on a 'qualitative mantle'.

## **Merger**

Hitherto the concept of the merger of an idea and its expression and the loss of copyright protection thereby has been a concept of American rather than Australian jurisprudence. It seems that in this regard, however, the two countries are moving closer together. On a number of occasions Jenkinson J seems to accept that the concept has relevance in this country. In finding that there is no infringement by the Error Text he states that 'the idea is inseparable from its function'.

## **Interoperability in Australia**

In choosing to construct his compression table in a particular way, it can be persuasively argued that Dr Bennett chose to follow the same methodology (not copyrightable in itself) and the same expression necessarily follows. If I know that someone has constructed a list of all the members of the UN ranked in accordance with the number of staff they have permanently in New York and I choose to construct my own list on the same basis, I should end up with a list which is the same as the original list. Such a list would attract copyright protection (even after *Feist v Rural Telephone Service Co Inc* (1991) 20 IPR 129 I believe there would be sufficient originality under US law for instance). However, it could not be

said that my list infringed the copyright in the original, even though I deliberately set out to use the same method to develop my list.

With respect, it is submitted that the present decision has made the same mistake as was made in *AutoDesk v Dyason*. The result in that case means that there is no legitimate method of reverse engineering the 128n bits which constituted the key at the heart of that case. This decision effectively grants DataFlex a monopoly in the compression table they have chosen.

Suppose that one had some DataFlex files which had been compressed. If the compression method was known and there was no copy of DataFlex, one might construct a compression table so that they could be decompressed. The files might then be translated to another convenient form. Based on his reasoning in this case, Jenkinson J would have to find the new compression table an infringement of the copyright in the DataFlex compression table. In the judgment, his Honour found that the infringement of copyright stems from 'the desire of Dr Bennett for the compatibility he achieved by reproduction of the DataFlex table.'

Similar reasoning could apply to any piece of code which is deliberately written to be compatible with another. Any printer which is designed to be compatible with another system must duplicate certain protocols so that it can operate. There is no reason why these protocols do not infringe copyright in the original following this decision.

This finding must make reverse engineers and potential reverse engineers in Australia very wary. It is not an overstatement to describe this decision, coming on top of *AutoDesk v Dyason* as a comprehensive defeat for interoperability in Australia. Copyright owners now have two decisions which can be used to stop compatible systems being produced. Under this decision it is not permissible to write a program that produces files which are compatible with those produced by another



program or to duplicate the functionality of a compression table. The fact that one is allowed to virtually copy the error messages of another program is cold comfort indeed.

It would have been straightforward to find an infringement in the compilation of the 192 words in the two languages which are the same. His Honour chose not to do so, although an appeal court might take a different view.

An analysis of the 192 words as a compilation would have created an excellent opportunity to explore the proper limits of reverse engineering. In conversation with the writer Dr Bennett has claimed that he created PFXplus so that there was the least possible duplication of material from DataFlex and still provide compatibility; the 192 words were the least he needed for the desired interoperability.

Had Data Access a monopoly right to those 192 words? Given that use of the same words is necessary for compatibility of the files produced, is there a legitimate way to reverse engineer those 192 words?

There are clearly a number of public policy questions involved as well as legal ones. How far should interoperability be encouraged or at least tolerated? Presumably it would have been possible to produce an inoperable program - that is one which could use data produced by DataFlex - without using those 192 words. Is it only in the attempt to create a truly compatible program where files can be used no matter which program produced them that the line of infringement is crossed?

Do 4GL's create special problems in this regard? There is now no question of infringement arising from the fact that word processing programs can read files created by rival programs. Should the fact that 4GL's can only achieve this level of compatibility by reproducing a list of reserved words deny users access to the same level of compatibility enjoyed in other areas?

It might be that an appeal will provide some answers.

### **Conclusion**

One of the curiosities of the case is that Data Access were represented in the case by Julian Burnside QC who unsuccessfully represented the Respondents in *AutoDesk v Dyason*.

It is difficult to reconcile a number of the findings made by his Honour. On the one hand there is copyright in a single word and in choosing a particular format for the compression table, an infringing copy was made. On the other hand the great similarity between the two error texts did not lead to an infringement.

Overall, however, it would probably have been surprising if a program which not only looked the same to a user but actually produces compatible files did not infringe the copyright in the original program. It would have been straightforward to find an infringement in the compilation of the 192 words in the two languages which are the same. It may be that an appeal bench will come to the same conclusion as the first instance judge but for this different reason. If so, we can only speculate on the outcome if the Respondents had chosen to construct a similar but different language which provided all the same elements of compatibility.

The very wide definition of 'computer program' which now includes a single word has the advantage of allowing a court to consider the question of copyright in a computer program without getting caught up in definitional squabbles. However, in finding that each word can be both a computer program and an individual copyright work Jenkinson J has dealt another huge blow to open systems in this country. It seems that the smaller the amount of code which constitutes a program the easier it is to find a substantial reproduction of it.

His decision in relation to the compression table may have an even more far reaching effect; it leaves a

reverse engineer no room to manoeuvre because of the finding that the infringement flows directly from the desire to produce a compatible system.

The finding of his Honour is consistent with the finding in *AutoDesk v Dyason*. The decision goes further than the decision of the High Court, for example in implying that a part of a computer program must be essential before it can be substantial, but stays within the framework laid down by that case. Whilst only a first instance decision, the case will therefore provide a precedent unless and until there is a successful appeal. Because of the different facts involved, it is also true that an appeal court could find for the Respondents without having to have *AutoDesk* overturned.

The difference in the result of this case on the one hand, and similar cases in the US, Canada and the UK on the other, could hardly be more marked. It seems likely that only legislative change, as is contemplated by the Copyright Law Review Committee, will bring Australia into step with the rest of the world.

It may well be that any changes which are made to the *Copyright Act* would not alter the decision in this case but changes are certainly needed so that this case is not used as a precedent to stifle legitimate open systems endeavours.

*James FitzSimons*  
*Clayton Utz*  
Ph +61 2 353 4199  
Fx +61 2 251 7832