Bond University
ePublications@bond

High Court Review | Faculty of Law

1-1-1998

Data Access v PowerFlex

William Van Caenegem
Bond University, William_Van_Caenegem@bond.edu.au

Follow this and additional works at: http://epublications.bond.edu.au/hcourt

Part of the Intellectual Property Commons

Recommended Citation
William Van Caenegem. (1998) "Data Access v PowerFlex" ,, .

http://epublications.bond.edu.au/hcourt/14

This Journal Article is brought to you by the Faculty of Law at ePublications@bond. It has been accepted for inclusion in High Court Review by an authorized administrator of ePublications@bond. For more information, please contact Bond University's Repository Coordinator.

Case Note: *Data Access v PowerFlex*

by William van Caenegem[1]

Headings in this case note:

1. Introduction
2. Computer programs and the concepts and principles of copyright law
3. Difficulties with the 1984 amendments relating to computer programs
4. Is a single word in a programming language "substantial" enough, and can it be "original'?
5. Some odd results of copyright in program words?
6. A tentative proposal for legislative change: split computer programs up between Part III and Part IV.

---

1. Introduction

{1} The issue in this case is relatively straightforward: does a single word in the DataFlex language amount to 'a set of instructions' in the sense of that expression in the definition of 'computer program' in s 10 of the *Copyright Act* 1968 (Cth)? In answering this question, we enter a well-worn and complex debate about protecting the rights of authors and subsequent owners of computer programs by classifying the latter as "literary works" in Part III of the *Copyright Act* 1968. [2]

{2} Resolution of the question is not in truth simple. Difficulties encountered are due to two factors: first, granting computer programs the status of copyrighted literary works causes conceptual stress, given the nature of other protected subject matter and the broad principles of copyright law; and secondly, the hasty introduction of amendments that inserted computer programs into the *Copyright Act* 1968 resulted in considerable uncertainty.[3] The inclusion of computer programs as literary works in Part III may well have resolved the immediate need, which resulted from the *Apple* case,[4] for some form of protection, but it also created a new set of difficulties, most of which have not been resolved by the courts or addressed by the Commonwealth Parliament.[5]

2. Computer programs and the concepts and principles of copyright law

{3} Some of the difficulties encountered in attempting to define a computer program in copyright terms are due to the irreconcilable tension between, on the one hand, the nature of the subject matter, and on the other hand, the conceptual basis of statutory copyright law and the principles employed in its application in disputed cases. A computer program is at once a functional and technical thing (i.e. an integral part of an operating or functioning machine, being a computer or any other machine that operates with a digital brain), and, at least at some stages of its development, also an expression in a language (a series of symbols with

meaning to at least some humans).[6] A computer program is thus an unusual amalgam: literary by conception but functional in operation.

{4} It is thus also fundamentally different from any other works in Part III of the Act. Such principles as the "idea-expression dichotomy" used by the courts to circumscribe infringement, the originality requirement for subsistence, and a qualitative rather than quantitative approach to the definition of a substantial part, are difficult to apply to material that is functional, rather than either informative or entertaining.[7] This is not to say that copyright law has been ineffective in giving computer programmers a level of safety against rip-offs, merely that it is difficult to apply some of the broader principles to such different subject matter, and consequently a need arises to develop principles specific to computer program copyright. This need neither creates greater certainty or simplicity in copyright law (one of the stated desires and aims of legislators in recent times), nor does it advance conceptual clarity with regards to other works or subject matter within the Act.

3. Difficulties with the 1984 amendments relating to computer programs

{5} Be that as it may, computer programs *are* literary works and we are thus confronted with a practical difficulty that must be resolved as effectively as possible. Unfortunately, the inclusion of computer programs in the Act was not only conceptually flawed, but also badly executed. *PowerFlex* illustrates shortcomings that had become apparent in *Autodesk Inc v Dyason*,[8] and that have been commented on, for instance, in the report on Computer Programs published by the Copyright Law Review Committee in 1995.[9]

{6} By virtue of s 10 of the Act, a computer program is a "set of instructions" (etc...). The term "set" is not further defined, nor is the term "instructions". "Set" implies plurality: to make up a set, at least 2 items are required, but a set can also consist of 200 or 2000, or any other figure above 2. A set is also a group of items that are not necessarily identical, but have some definable similarity and inter-relationship. "Set" is commonly used for objects,[10] but is also used for symbols (an accountant might refer to "a fine set of figures").

{7} As a consequence, what *amounts to* a "computer program" under the Act, is, as long as there are 2 or more instructions, in terms of size, quite open. It could be a whole package (e.g. Windows 98), or only a part of the package (that part of the Windows 98 code that causes the appearance of an icon on the screen for instance, or the Multimedia function in Windows 98). But might it also be far less?

{8} By virtue of s 10, a "set" may be expressed in object or source code.[11] Hence one is forced to conclude that any sequence of *2 or more "digits"*, as long as it is intended to cause a computer to perform a particular function, is a computer program. Thus a single word in source code (i.e. a programming rather than an operational language) if it represents a sequence of 2 or more digits in object code (i.e. an operational language) that is intended to cause a computer to perform a particular function, is a computer program. The words "a particular function" limit a computer program to a sequence that makes a computer do something; but the definition does not limit a computer program to something that in practice allows a user to do something with or on a computer.[12]

4. Is a single word in a programming language "substantial" enough, and can it be "original'?

{9} Of course, that a putative computer program accords with the statutory definition of a computer program, is not in itself sufficient: it must also *amount to* a literary *work.* In *Exxon* the term "work" was interpreted as requiring some substance, i.e. something too small or insignificant could not amount to a work correctly so called.[13] It was also said in that case that a *literary* work was some expression which conferred some literary enjoyment or information. However, the inclusion itself of computer programs in the Act has cast that requirement into doubt: a computer program obviously - by its own definition in s 10 - does not have the purpose of informing or entertaining, but of giving operational instructions to a computer, of which it forms an integral part. So we can probably ignore the literary enjoyment or information requirement, at least in this context; in any case it has often been argued that that requirement mentioned in *Exxon* set too restrictive a threshold test anyway. That leaves us with the requirement of some minimum size.[14] To some degree the emphasis on this requirement, when it comes to copyright in single words is inspired by concerns about the overlap with Trade Marks law, as was the case in *Exxon*. But as Ricketson points out: "The denial of protection in such cases seems to be based on a *de minimis* rule, although there is also the consideration that copyright in a phrase or single word may intrude too much on the rights of the public." Can that concern be discounted in the context of computer programs? The answer may be that, since a word in a computer source code can in fact represent, in object code, a whole series, sequence, or set of instructions for a computer, size, in the context of computer programs, may not always mean much. It would therefore be artificial to impose an arbitrary restriction by arguing from *Exxon* type cases. There seems little basis to argue that a single word in a computer language, as long as it is original, cannot amount to a computer program.

{10} Where a word is not part of a known language, i.e. is invented by the author, there seems little doubt that it may be original in copyright terms. But what if the word is a known word to which some specific new meaning - relevant in terms of computer programming - is ascribed? Is it simply a question of fact and degree, and is it therefore not excluded entirely that a known word used in a new sense might amount to an original work? In the *DataFlex language* case, words were either modified or had a suffix added, with the inclusion of symbols such as a full stop. However, their meaning was reasonably discernible even without specialist knowledge (one of the advantages of a good source code language). It is certainly arguable that they are sufficiently original to amount to protected works.

5. Some odd results of copyright in program words?

{11} Some further points concerning the 1984 amendments are of at least passing significance to the debate concerning the DataFlex language. First, the definition of "adaptation" in the Act was amended to include versions of a computer program, but with no explanation of what this means. If it does mean an improved edition of a program (Windows 98 as a new version of Windows 95, for instance), why did it need to be included, since this would surely amount to a reproduction? What makes the answer even more elusive is that the definition actually provides that a version may be in the same language, code or notation (thus adaptation does not merely amount to translation) "not being a reproduction of the work".[15] If what is meant is something that is *a functional equivalent*, but quite different in

its expression in object (machine readable) code, should such a broad right have been granted to the copyright owner?[16] Is it compatible with the other computer program related provisions in the Act? In terms of the DataFlex issue, what is a version of a computer program consisting of a single word? Any word that performs the same function as the original word? The word "version", unclear as it is in general, is not well adapted to this circumstance.

{12} Secondly, a literary work is defined in s 10 of the Act as including a compilation of computer programs. What effect does this have? What is a compilation of programs rather than a program per se? The purpose of the inclusion of compilations may well be to indicate that a computer program can be sufficiently original to amount to an original work, even though it is composed of pre-existing elements (or programs, or sub-programs). It is the skill, judgment and effort of an authorial kind put into compiling the programs that may amount to sufficient originality. In the context of this case, one could thus argue that a compilation of pre-existing programming language words can amount to an original computer program and thus be protected by copyright. That leaves scope for the work of a computer programmer who uses the language that consists of original source code words, to produce an end result that is protected by copyright. His or her investment in time and effort is thus protected, at least to a degree. However, in the course of producing such a compilation without the license of the owner of the copyright in the words, the compiler would arguable infringe the Act.

{13} Some of the odd results flowing from copyright protection of source code words further highlight the deficiencies of the 1984 amendments. But they do not amount to a sufficient argument to conclude that the provisions of the Act (as it stands since those amendments were made), do not permit of copyright protection for single words in a computer programming language.

6. A tentative proposal for legislative change: split computer programs up between Part III and Part IV.

{14} The general conclusion derived from the foregoing discussion is that *arguably*, on the basis of the terms and provisions of the Act, a word in the DataFlex language (and in any other computer language) can amount to a computer program, all other conditions for copyright protection being fulfilled. It is well known that that conclusion has caused consternation in computing circles. Unfortunately, because of legislative inactivity, the task of finding a way out has come to rest with the High Court. Anxiety about the practical implications of the Court confirming the judgment at first instance, has a rational basis. The effect on inter-operability, and on effective competition and product development in the computer industry, could be negative.

{15} In response, one might argue, *ex absurdo*, that it is a ridiculous thing to say that one word in a source code could amount to a computer program; or that this was clearly not the intention of the legislature. But that would be to ignore the actual terms of the 1984 amendments, whether or not those terms amounted to an inadequate translation of the legislators intent. Alternatively, one may argue that the whole definition was conceived to protect only computer programs that are complete or that can actually be operated by a person using a computer. It is not possible to operate only a small part extracted from the whole Windows 98 package; one must have the whole package to operate a PC; the parts

operate (in the sense of interacting with a human operator of a computer or digital machine) only within the structure or framework of the whole, and therefore only the whole can *actually* cause a computer to perform a certain function, not some independent part. However, the Act requires that the expression must be *intended to* cause a computer to perform a function. That it in itself, for practical reasons, is incapable of performing this function, without conjunction with other programs, may thus be irrelevant since it is the intended purpose that counts. Furthermore, there is no doubt that the Act protects a computer program in source as well as in object code. Source code does not *actually* cause a computer to operate; this occurs only after translation into object code.

{16} All this leads to the inescapable conclusion, as if it needed to be said again, that amendment of the Act, as recommended by the CLRC report on Computer Programs and by many commentators, is long overdue. Easy to say, but how is this best done? In my opinion, the recommendations of the CLRC do not address the fundamental "round peg into a square hole" problem of the inclusion of computer programs in Part III literary works, although they may well address the technical deficiencies - of which there are many - with the present provisions. So, in the furtherance of debate on this issue, a more radical proposal may be required, taking into account that our international obligations, and foreign models and expectations, now have made *sui generis* legislation an improbable option.

{17} A possible solution may be to split up computer programs into two categories: human readable computer programs, i.e. programs in some human readable source code (something humans can make sense of, even if only highly skilled humans); and computer readable computer programs, i.e. programs in binary or object code or stored as electrical impulses.

{18} The former would remain categorised as literary works under Part III of the Act; the latter would be protected as subject-matter other than works in Part IV of the Act. This would be apt because computer programs are only truly authored in human readable form; and since machine readable programs would be derived from, "neighbouring to" Part III literary works, i.e. the programs in human readable code. Digitally recorded computer programs are also a *technological product*, that is manufactured, as are sound recordings, films, and broadcasts. Underlying computer programs as literary works in Part III would remain protected. Infringement of Part III programs might be approached either on the basis of comparing human readable codes, something more easily akin to assessing infringement by unlicensed reproduction in traditional literary works, or by reproduction in binary code, akin to the reproduction of literary works in cine films. Infringement of Part IV programs would be limited, as with other subject matter, to direct and literal copying by mechanical or electronic means.

Endnotes:

1. Associate Professor of Law, Bond University

2. See for instance Copyright Law Review Committee ("CLRC"). *Report on Computer Software Protection* (1995).

3. See the *Copyright Amendment Act 1984* (Cth.)

4. See *Apple Computer Inc v Computer Edge Pty Ltd* (1983) 50 ALR 1; 1 IPR 353 (at first instance); *Computer Edge Pty Ltd v Apple Computer Inc* (1986) 161 CLR 171; 6 IPR 1.

5. The recommendations of the CLRC (above n 2) have not yet been acted upon.

6. One can easily quibble with this: is a symbol with meaning or relevance only to a machine (algorithms in object code) a language?

7. Note for instance the difficulty of finding a balance between 'look and feel' - type protection, and literalist protection, as was, maybe somewhat artificially resolved in *Autodesk v Dyason (No 1)* (1992) 173 CLR 330; 22 IPR 163 by insisting that an operationally significant part equated to a qualitatively important part, rather than the level of originality, size or structural importance within the context of the overall practical purpose (i.e. Computer Aided Design) of the program.

8. See above n 7.

9. See above n 2.

10. Cf the definition of a 'set of articles' in the *Designs Act* 1906 (Cth), s 4.

11. See s 10 definition of computer program: 'either directly or after any of the following: (a) conversion to another language, code or notation.'

12. It should be added that the inclusion of the terms '(whether with or without related information)', makes things even less clear. Does it mean that a non-functional part of a computer program can in fact by itself amount to a computer program, as suggested in *Autodesk v Dyason*. If so, that again seems an unsatisfactory outcome.

13. *Exxon Corporation v Exxon Insurance Consultants International Ltd* [1982] 1 Ch 119.

14. See Ricketson S, *The Law of Intellectual Property,* LBC 1984 at [5.61]: "Copyright will not usually be held to subsist in works which are 'insubstantial'." (p 110).

15. That this form of adaptation is not a reproduction brings it out of line with all other forms of adaptations, which are generally regarded as amounting as well to reproductions, although they may be, when appraised in terms of literal similarity, quite different (in the sense that a French translation of Patrick White's *The Tree of Man* would appear quite different from the

original but would still be seen as a non-literal reproduction), or, as is sometimes said, not 'objectively similar'. A further example is the reproduction of a literary work in a cine film.

16. In *Autodesk v Dyason* (see above n 7) the High Court stressed that approaching infringement, as the trial judge had done, in terms of functional equivalents, is not correct; yet they were considering infringement of the reproduction right, not, it seems, the adaptation right, the significance of which remains largely untested. The suggestion that an adaptation in terms of computer programs could be interpreted as a functional equivalent, although it may excite some copyright owners, is a little mischievous: see for a more orthodox explanation concerning the confusion between reproduction and adaptation in the context of translation from source to object code and vice versa, CLRC *Report*, above n 1, at [6.63] - [6.87] and [9.51] et seq.