Database case

Linklaters & Alliance Intellectual Property News

EU

In one of the first decisions since implementation of the Database Directive (96/9/EC), *British Horse Racing Board v William Hill,* the defendant's use of horse racing information supplied to it by a third party amounted to a breach of the claimant's database rights in that information under Article 7 of the Directive.

The British Horse Racing Board (BHB) is the governing authority for the British horse racing industry and part of its work involves the maintenance of a database comprising details of races to be run in Great Britain which BHB made available under licence to a third party, Satellite Information Services (SIS), for onward transmission to its own subscribers by means of a raw data feed. William Hill had used information from the SIS raw data feed in its betting shops without objection from BHB. William Hill had also recently embarked upon the provision of betting services over the Internet using information derived from the SIS data. It was common ground that SIS had no right to sub-license the use of the pre-race information by William Hill. BHB claimed that William Hill was in breach of BHB's database right by extracting or re-utilising a substantial part of the contents database contrary to Article 7(1) of the Directive; or by repeated and systematic extraction or re-utilisation of insubstantial parts of the database contrary to Article 7(5) of the Directive. William Hill argued that database right protected the form of the database whereas it had merely used the information contained in it which was not individually protected.

It was held that:

- there was infringement within Article 7(1) of the Directive. Copyright protected the form of a database while database right protected the investment in it. An infringer took advantage of that investment if he made use of the accuracy of the data, not because he took it in a particular form. William Hill's action in taking the SIS information and loading it on its own computers was an unlicensed "extraction" of a substantial part within Article 792) and making it available on its website was a "re-utilisation"; and
- as to Article 7(5), the BHB database was to be regarded as one database in a state of constant revision. William Hill's borrowing from it day-to-day was a repeated and systematic extraction and re-utilisation of parts of its contents. Williams Hill's activities clearly undermined a significant part of the BHB's exploitation of the database and unreasonably prejudiced its interests.

(This article was supplied courtesy of Linklaters & Alliance Intellectual Property News Issue 15, March 2001)

Managing intellectual property issues in the software development process*

Megan MacGregor, Corrs Chambers Westgarth

Megan joined Corrs Chambers Westgarth as a Senior Associate from a leading firm of solicitors in Scotland. She was admitted as a solicitor in Scotland in 1991 after training with Macroberts in Glasgow. She has lectured at Napier University in Edinburgh and introduced courses in Intellectual Property, Information Technology and Media Law which are her principal areas of practice.

Introduction

The development of computer software represents a significant investment for most companies. The research, design and development of new software applications involve the outlay of considerable time and expense. Today specialist software houses and consultants develop most software on a consultancy basis.

In some cases the company commissioning the software will assume that it owns all or part intellectual property rights in the materials that the software house creates. Ownership of intellectual property will often be discussed at the contract negotiation stage and a draft heads of agreement or software development agreement may be produced to attempt to set out the parties' rights. Numerous faxes, letters and e-mails may be exchanged on the topic.

Ultimately, many software agreements are destined to remain unsigned because key issues, including ownership of, or warranties in relation to intellectual property in the software, cannot be agreed upon in advance. This can cause problems at a later stage when software developed externally is needed to continue to provide services to customers particularly in the context of a merger or acquisition of the client company's business. Companies should therefore take steps to manage, identify and confirm ownership of key intellectual property holdings on a regular basis.

Ownership of Intellectual Property Rights

One of the most important functions of any software development agreement is to establish who owns the intellectual property rights (**"IPR"**) in any software that is created by an independent software house.

In most legal systems including Australia, the ownership of IPR in a software program belongs to the author of that program and not to the person who commissions and pays for the development work unless the creator has signed a written assignment of the IPR.

In an ideal world, the ownership of all IPR in software to be developed by independent contractors would be agreed from the outset. Formal assignments or confirmatory assignments would have been executed where appropriate. In the real world, of course, this does not happen. To make matters worse, in many situations the company commissioning software development does not ensure that it obtains the software house's consent to use, and the right to sub-licence other users to use, the IPR in the software created. A client company may have used software for years only to find that the software house that created it has been sued for copyright infringement. This will almost certainly prevent the software house from continuing to offer maintenance and support to its client. The third party may then decide to raise infringement proceedings against all users of the software, including the client, in order to obtain injunctions. While it will be possible to negotiate a licence with the third party to continue to use the infringing software, this may be on terms that are unfavourable to the user.

In any negotiations relating to software development the parties should consider ownership of IPR and the extent of any licence to use the IPR in the software as a material issue.

Identifying Intellectual Property Rights in software development

In summary the IPR most relevant in software development are:

- Copyright (protects original works of authorship from unauthorised copying);
- Moral rights (newly introduced to protect the honour and reputation of the authors of software, where these are individuals);
- Patents (protect new, useful and non-obvious inventions and processes);
- Confidential information (protects valuable proprietary information that is not in the public domain);
- Trade marks (protect the goodwill in a product or services by indicating the source of that product or services associated with the trade mark);

Dealing with copyright in software

A number of different copyright works will be produced during the development of a computer program. The software development process will normally begin with the generation of a broad specification of the general intended purposes and functions of the software. This will be followed by a more detailed specification setting out the particular functions and operations of the program. Both these stages will involve detailed discussions between the software house and its client. The specifications may then be set out in the form of a written document or a flow chart together with pictorial 'screen dumps' to show the visual appearance of the intended program. Each document in the development process will be treated as a copyright work in its own right. Written documents, no matter how simplistic, will be treated as literary works. Similarly, sequence and state chart diagrams and pictures of screen displays will all be treated as artistic works provided these are original.

The computer software program itself will be protected as a literary work under the *Copyright Act 1968* **("the Act")**. The Act was recently amended[1]. A 'computer program' is now defined by Section 10(1) of the Act as being *"a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result"*.

A new, supplementary definition of computer program has also been introduced in Section 47A of the Act for the purpose of establishing the extent to which certain acts will not infringe copyright in computer programs. Section 47AB provides that:

> *"a computer program includes any literary work that is:*
>
> *(a) incorporated in or associated with, a computer program; and*
>
> *(b) essential to the effective operation of a function of that computer program."*

In the past, the treatment of software as a literary work has created interpretation problems for the courts. While there was traditionally no difficulty in treating the written parts of a program (ie the source code) as a copyright, literary work, the courts could not treat the non-written parts of software programs (the object code) as a copyright work[2]. Some writers have expressed the opinion that the new definition of computer program in the Act will allow **all** parts of a software program to be protected by copyright – both the source code and the object code[3]. There will be no way to ascertain whether this is the case until a court provides a ruling on the extent of protection provided by the definition in Section 10(1) and its relationship with the definition in Section 47AB.

To avoid doubt, therefore, parties should make it quite clear in the software development agreement who will own the copyright in the various materials produced in the development process.

Computer software and moral rights

Moral rights are a new concept in Australia, although long recognised in Europe. The latest revision of the Berne Convention[4] included reference to the moral rights of authors. Australia was bound as a signatory to the Berne, to introduce moral rights into Australian law.

Under recent amendments[5] to the Act, the following moral rights were introduced in Australian law on 21 December 2000:

- the right of attribution of authorship;
- the right not to have authorship falsely attributed; and
- the right of integrity of authorship.

Moral rights are separate and distinct from ownership of copyright. Copyright law in continental Europe has always divided the author's rights in economic rights and moral rights.

Economic rights allow an author to earn money, moral rights allow an author to protect his or her reputation. Only authors who are individuals have moral rights[6].

Moral rights remain with the *author* of a piece of software, even where the software, or the copyright in the software has been sold or assigned; regardless of whether the author created the software in the employ of someone else, or created it under contract, or otherwise.

Accordingly, if the owner of the copyright in the software or one of its licensees subsequently infringes the author's moral rights, the author can raise proceedings against the infringing party. While it is probably unlikely that the courts would award an author substantial damages for infringement, an author could cause problems for licensors and licensees alike by seeking interim injunctive relief to prevent use of infringing software.

Interestingly, the Act has deviated from the lead set down by most other countries. In other countries, the UK for example, the creators of computer programs are excluded from the protection afforded by moral rights[7]. Under the Act, computer programs are treated as any other literary work and the individual creators of software can claim the protection afforded by moral rights.

(a) The attribution right

The attribution right entitles an author to be identified in a manner likely to bring his or her identity to the attention of a person seeing or using the work in question. The author has the right to be identified on every copy.

In each case the identification must be clear, visible and reasonably prominent. The author may require that a pseudonym, initials or some other particular form of identification identify him or her. For example, in a software program the authorship notice should be visible to persons using the program. The notice should therefore be used in labels affixed to the software and also on screens within the program where a notice can be generated.

If the author does not specify a form of identification (or pseudonym or initials), any reasonable form of identification may be use

(b) The right to object to false attribution

Authors may be angered or suffer loss if a work of poor quality is falsely attributed to them. While the law of defamation may give some remedy to an author if a false statement is made, the necessary evidence of defamation may not exist. The right provides a useful remedy where the author is not a professional writer and could not therefore recover damages for the loss of goodwill in a passing off action.

(c) The right of integrity of authorship

The right of integrity gives an author the right to object to any derogatory treatment of his or her work. This right extends to any treatment of the work that amounts to a distortion, modification or mutilation of the work which is prejudicial to the honour or reputation of the author. The right of integrity would prevent a software house from reproducing or publishing a consultant's program that had been altered without authority, if that alteration amounted to derogatory treatment.

Dealing with moral rights in software development contracts

Traditionally, lawyers acting for employers and principals have advised employers and principals to require employees and consultants and contractors to contractually waive their moral rights. This is no longer possible under the Act. Moral rights can neither be waived nor assigned. The moral rights set out in the Act may, however, be circumvented to a certain extent if the author of the copyright work specifically consents in writing to acts or omissions that would otherwise infringe his/her moral rights.

Therefore, with respect to agreements with consultants or independent contractors, software houses should insert a clause requiring the consultant or independent contractor to:

(a) grant consent (if an individual); or

(b) procure the consent of the author of the work (if a company) to carrying out specified acts or omissions by the principal. For example, the author of a work may consent to the principal reproducing the work without attribution of authorship.

Employees should also expressly grant consent to their employers to do, or omit to do, any acts in relation to all copyright works created by the employee in the course of his or her employment. Where, for example, an employee prepares a handbook or instructional manual to accompany a program in the course of his or her employment, an appropriate moral rights consent should be obtained.

Contractual consent as a defence to infringement of moral rights

Software development contracts should contain warranties that the authors of software grant consent to reproduce, transmit, publish, modify or alter the work - without attribution of authorship; bearing false attribution; and to modify or alter the work even if this constitutes "derogatory treatment" of the work which is prejudicial to the author's honour or reputation. With respect to agreements with consultants or independent contractors, it is prudent to insert a clause requiring the consultant or independent contractor to:

(a) grant consent (if an individual); or

(b) procure the consent of the author of the work (if a company) to do **specified** acts or omissions by the principal. (For example, the author of a work may consent to the principal reproducing the work without attribution of authorship.)

Software development contracts should impose an obligation on the software house to obtain express consent from its employees to do, or omit to do, any acts in relation to all copyright works created by its employee in the course of employment. Consents may be invalidated by duress or false or misleading statements and accordingly, express consents should ideally contain a statement that the consent has not been obtained under such compromising circumstances. It should be noted that where external evidence exists that a software house has demanded moral rights consents from its employees and contributors, this may negate the consent[8].

Dealing with the disclosure of confidential information

During the planning and development stages of software development, a client will need to discuss the product with the software house. If the product is likely to be commercially valuable it will generate interest and can be very easy for key information about a product or a client's plans for a product to be disclosed either deliberately or inadvertently to a competitor.

From the very start of the development process it is fundamental that all information about a new software product and the client's uses for the product (including pricing) be kept confidential. Confidentiality is critical to successful negotiation with industry partners and information should only be disclosed to or by a software house on a need- to-know basis. If details about a product become public, even on a limited basis, this can jeopardise a later patent application in respect of the product.

Whenever possible, information about products and underlying processes should be protected using a bespoke confidentiality agreement. Contracts should contain clauses that require the receiving party to maintain the confidential nature of the software and the methods of developing that software. The client should be placed under an obligation not to disclose or use the software except in accordance with the express terms of the contract and it is sensible to identify the name and version of the software being disclosed to avoid doubt. At all times, access to source code should be strictly limited.

Clients on the other hand should ensure that the software house, its employees and sub-contractors do not disclose (or use) its business methods, marketing concepts, technical information, customer/prospect details or pricing strategies.

Confidentiality agreements should be tailored to specific circumstances in individual software development situations. There is no such thing as a "standard" confidentiality agreement, particularly when the agreement is supposed to incorporate mutual obligations. The aim of a good confidentiality agreement is to clarify what information has been identified as confidential and by doing so to strengthen the protection that exists for confidential information at common law. The more precautions that have been taken to protect valuable information from being disclosed, the more likely it is that the courts will treat that information as confidential.

Software and business method patents

From recent publicity there seems to have been an international increase in the numbers of patents being applied for and granted for inventions involving software programs and business methods. This trend is not surprising bearing in mind that it has been argued that copyright does not provide sufficient protection to software[9]. In some circumstances software may qualify for patent protection. While computer software is not inherently patentable, the Australian Patents Office Manual of Practice and Procedure **("the manual")**[10] lists certain categories of subject matter that requires special consideration including computer software and business related software and plans. Such items are normally excluded because the subject matter cannot be said to be a manner of manufacture. If, however, a practical application can be established for computer software or a business scheme incorporated in software then a patent application may be successful[11].

The manual provides that computer software and arguably, by extension, business methods implemented in software programs may be patentable if it "has been implemented" in some way. If the subject matter is executable computer code that constitutes a novel method of producing a certain result, then a patent may be granted in respect of that program. For example, where the invention consists of computer software that allows a computer with a slow processor to process information as quickly as a computer with a faster processor this would be patentable[12]. The key is to draft the patent claims for the invention in terms of the apparatus or method that supports a particular, novel function.

On its website[13] the Australian Patent Office has posted a document titled "Patents for Schemes or Plans" that states:

> *"Essentially a patent may be granted for a scheme or plan where there is a means of putting the scheme or plan into effect. A scheme (including a business scheme) or plan, by itself, is not suitable for a patent because it does not specifically give rise to an artificially created end result of economic utility ... The patentability of the scheme or plan resides in artificially putting the scheme or plan into effect".*

Questions over patentability of software and business method patents were discussed in the recent US case *State Street Bank v Signature Financial Group*[14].

This case upheld a number of earlier US decisions finding that computer software was patentable if it could be claimed that the invention embodied in the software had a practical application.

Bearing in mind that patents represent valuable assets, the importance of maintaining confidentiality about the product throughout the software development process is a key issue.

Dealing with trade marks in software development

While not a protection for the content of a software program, trade marks have a valuable role to play in establishing goodwill in a software product name thereby adding value to a brand. Trade marks allow potential clients and customers to identify and distinguish a good software product from those produced by competitors. Trade marks can identify the origin of a product or serve as an indication of quality. The commercialisation of IPR is often heavily dependent on trade marks and branding issues.

Both software houses and clients should be sensitive to the fact that a product name may become an important asset in its own right. Accordingly it is always advisable to conduct a preliminary trade mark search in Australia before selecting a product name and ensuring that appropriate use is made of the trade mark throughout the life of the product.

It would do no harm for software houses to circulate trade mark usage Draft Guidelines to their employees and consultants as well as their clients to safeguard trade marks. This is a subject that can also be dealt with in the IPR licence included in the software development agreement. A client, for example, should be obliged to comply with any reasonable directions relating to the use (including the appearance) of the software house's trade marks.

Managing IPR in the software development process

Software houses should conduct thorough IPR audits throughout the development process to identify whether any external consultants worked with its employees to create software products. If so, a written agreement assigning all ownership rights in the IPR produced by the consultant should be obtained. The agreement should also contain a carefully drafted moral rights consent clause and a strict confidentiality undertaking. If the consultant refuses to sign an agreement or cannot be located afterwards, the software house should make efforts to identify and isolate what material the consultant contributed to the development project. Depending on the extent and nature of the contribution it may be possible to argue that the consultant did not contribute sufficient material to justify a claim of ownership of IPR in that material.

All employees who are key developers should from the commencement of their employment be asked to sign agreements assigning ownership of all inventions and copyright works to the employers. Employment contracts should also include a moral rights consent clause, confidentiality and non-compete undertakings. Such agreements can be helpful in avoiding disputes where an employee works from home or outside normal business hours. Such agreements should also contain a standard "boilerplate" clause requiring the employee to sign any other documents necessary to assign ownership in the IPR that he or she creates to the employer.

[*] This article is based on a paper delivered by the author and Kim Edwards, Senior Solicitor, Corrs Chambers Westgarth at *"Negotiating and Drafting Watertight IT Contracts"* organised by IES Conferences on 30 March 2001 in Sydney.

[1] By the Copyright Amendment (Digitc' Agenda) Act 2000 (Act No. 110 of 2000)

[2] ee Autodesk Inc. v Dyason (1989) 15 IPR 1; Dyason v Autodesk Inc (1990) 24 FCR 147 Data Access Corporation v Powerflex Services Ltd (1996) 63 FCR 336 (Data Access No. 1); Data Access Corporation v Powerflex Services Ltd (1997) 75 FCR 108 at 114 (Data Access No. 2); Data Access Corporation v Powerflex Services Ltd (1999) 45 IPR 353 (Data Access No. 3); Admar Computers Pty Ltd v Ezy Systems Pty Ltd (1997) AIPC 91-350

[3] See T Voon *"Revisiting Computer Program Copyright in Australia: Data Access Corporation v Powerflex Services Pty Ltd"* 2000 Australian Intellectual Property Journal Volume 11, 161; N Blackmore *"Debugging the Application· Copyright Protection for Software in Australia"* 1999 Australian Intellectual Property Journal Volume 10, 67.

[4] Article 6 *bis* Berne Convention for the protection of Literary and Artistic Works (Paris text 1971)

[5] Copyright Amendment (Moral Rights) Act 2000 (Act No. 159 of 2000)

[6] Section 190, Copyright Act 1968

[7] Section 81(1) of the Copyright, Designs and Patents Act 1988 (UK)

[8] Mike Holderness in his article *"Moral Rights and Authors Rights: The Keys to the Information Age"* (1998(1) The Journal of Information Law and Technology (JILT) http://elj.warwick.ac.uk/jilt/infosoc/98_1hold) relates the news of an anonymous fax to the National Union of Journalists in the UK reported that the media group VNU demanded that freelance writers sign forms waiving moral rights in order to receive payment for their work. Similar implicit threats will invalidate any consent given by authors/creators in Australia.

[9] J R Kuester *"As Software Patents Take Over, Expertise Is Key"* (1998) The National Law Journal B13 Accessible online at http://www.ljextra.com/practice/computer/0420softpat.html (Last accessed) 30 April 2001), N Stoianoff "Patenting Computer Software: An Australian Perspective" [1999] EIPR 500

[10] Posted online at http://www.ipaustralia.gov.au/patents/Manual/P_manual.htm (Last accessed 30 April 2001)

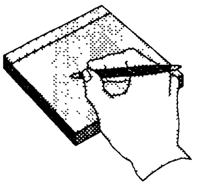
[11] See International Business Machines Corporation v The Commissioner of Patents (1991) 105 ALR 388

[12] See examples given by Tim Rickard in his article *"The Patentability of Computer Software in Europe"* SCL Magazine, October/November 2000 Volume 11 Issue 4

[13] http://www.ipaustralia.com.au

[14] 149 F.3d 1368, 1375 (Fed. Cir. 1998)

Contribute to the Journal!



The editors welcome contributions to the journal.

We encourage submission of articles, casenotes, reviews and comments on topics relating to computers and law.

The following are some topics you may be interested in submitting a piece on:

- casenote on any recent litigation in Australian and New Zealand courts;
- the internet, content regulation, jurisdictional and conflict of law issues;
- issues arising from computer contracting;
- e-commerce and related subjects such as financial services and securities dealing on the web;
- Privacy, consumer protection and security issues concerning the use of computer technology.

Please feel free to submit papers on topics of your choice that are of current public interest . The above suggestions are intended merely as "springboards".

Please refer to the "Notes for Contibutors" section for more details (page 51).