

LEGAL PROTECTION OF SOFTWARE

A recent EC Directive seeks to prevent illicit copying of computer programs (software) and to harmonise legal protection throughout the Community. However there is debate within the software industry on whether the Directive will stifle parts of the industry it is designed to protect.

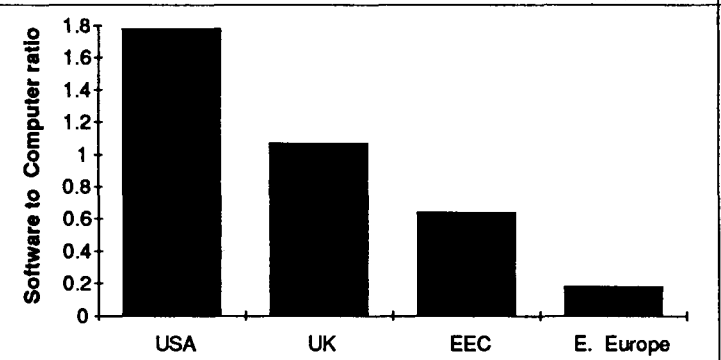
This Briefing Note examines the Directive's potential impact on the software industry and related issues.

COMPUTER SOFTWARE

Before a computer can be used, it must be loaded with a computer program which provides the instructions to carry out the tasks required. Computers are commonly referred to as 'hardware'; thus computer programs have become known as 'software'. As computers have become so widely used, demand has increased for software and associated support services (consultancy, system maintenance, training, etc.). In the UK alone, expenditure on software and services topped £4,200 million in 1990 and there are now more than 2,300 UK-based software suppliers. The most popular software items sell in their millions, and thousands of new products are introduced each year. In the EEC as a whole, the sales of software and related services are expected to outstrip those in the USA by the mid-90s.

Software is usually sold recorded on 'floppy' discs and can be copied onto blank discs by the computer (just as taped music can be copied onto blank tapes). Like any other written composition, software is protected by the laws of copyright. But whereas music can be copied for home use, software is sold under a licence agreement which specifies the conditions under which the program may be copied to a computer or separate disc. Wholesale copying of software by counterfeiters is clearly unlawful, but the licence agreement usually specifies that each software package can be used on *one* computer only - thus use on a second or third computer, let alone deliberate duplication onto floppy discs for dissemination (e.g. around the office) is unauthorised and illegal. The latter kind of copying is widespread: e.g. a MORI poll of senior UK managers in 1990 found that 55% of them made extra copies of commercially available software, many being unaware that their practice was illegal.

Figure 1 RATIO OF COMPUTER PROGRAMS LICENSED FOR EACH COMPUTER SOLD IN A NUMBER OF COUNTRIES (1989)



The extent of copying in a number of countries can be estimated by looking at the sales ratio of software to personal computers (see Figure 1). Computers always require at least one and often several items of software for each machine, and thus a software-to-computer sales ratio well above 1 would be expected in the absence of copying. The fact that the ratio in Figure 1 is often *less* than one indicates that much software has been acquired without purchase.

Concern over illegal copying has led to the formation of industry associations (e.g. the USA's 'Business Software Alliance', the 'Software Action Group Europe' (SAGE) with 1,400 members, and the 'Federation Against Software Theft' (FAST) in the UK). FAST was formed in 1984 with the encouragement of the Home Office and has around 150 members from the software industry and legal firms. FAST estimates that illegal reproduction of software, whether deliberate or in ignorance of the law, costs the UK software industry £300M per year. Worldwide annual losses as a result of software piracy are estimated at £6,000 M, while job losses in the EEC and USA may run into the hundreds of thousands.

Industry groups like FAST argue that all copying of software is theft, and support greater legal protection of software and better enforcement of current regulations. There are however, a number of 'grey' areas within the software industry where copying is currently part of legitimate and essential activities, i.e.

Software Development. A software company will often wish its standard products to be useable on as wide a range of computers as feasible. Each computer has its own internal software which controls the basic operation of the machine - for example, the instructions which control how information is presented on the screen. For an external piece of software to work properly, information is needed on how the internal software works for each computer. In addition, many items of software developed today are designed as



DECODING SOFTWARE

Just as an author writes a novel in a natural language, a computer programmer uses a language to specify a sequence of instructions for the computer to follow. This list of commands - the *source code* - contains English words such as 'PRINT' or 'STORE' and would be readily understood by a fellow programmer. However, in order for a computer to follow instructions, the source code must be translated or *compiled* into a machine-readable form, or *object code*. This version of the program consists of a string of binary digits (i.e. '1's and '0's'), and it is this string that is stored on magnetic disc. The object code is effectively incomprehensible, but it is possible with considerable effort for an expert programmer to translate the object code back to source code and thence to understand how the program operates.

This reverse process is called *decompilation* and may be necessary in order to adapt software for certain tasks; for instance, if information is required on how another program controls the *interfaces* between various parts of the computer (e.g. the memory, central processor, keyboard, visual display units and disc drives), as well as with any interfaces with external facilities (e.g. other computers on a network).

add-on products to increase the usefulness of existing software; thus a special graphics package might be a valuable addition to a word processing program written by another software company.

Interoperable Systems. To improve communication in modern business practice, professional users tend to cable their computers together in a network (currently, 30% of small computers for business use are operated in local networks; this is expected to rise to 60% by 1993). With computers of different makes operating different programs, special software is required to ensure that all the computers can communicate with each other; a concept known as *interoperability*. Again, writing the appropriate 'connecting' software requires an understanding of the internal software of each different machine type.

For these and other purposes, programmers need to know how to *interface* their product (see **Box**) with the original machines or programs. The necessary information on interfaces is not always available from the vendor of the computer or original software, in which case a programmer has to deduce the interfaces by *decompilation* of the original program (see **Box**).

Thus some computer specialists and companies see a blanket prohibition on copying as placing a legal restraint on their own currently legitimate activities. Companies which share these concerns have formed the 'European Committee for Interoperable Systems' (ECIS) which is comprised of some 60 firms involved in hardware and software development. They see a need to protect their ability to compete in the software and hardware market, and regard regulations which restrict software development methods as an attempt by large companies to neutralise the activities of (often smaller) competitors wishing to create interoperable systems. Larger companies say these fears are groundless and that European Community laws on competition should prevent companies from exploiting strengthened copyright protection in such a way.

SOFTWARE PROTECTION LAW

The Copyright, Designs and Patents (CDP) Act of 1988, which subsumed the Copyright (Computer Software) Amendment Act of 1985, provides specific protection in the UK for software. In addition, copying of software might be regarded as 'unauthorised access to computers' and therefore subject to the provisions of the Computer Misuse Act (1990). In cases of blatant counterfeiting, other legislation such as the Trades Description Act (1968), can also be invoked.

Enforcement of copyright is largely a matter for copyright owners to control themselves via civil remedies. However, the CDP Act also contains penalties of unlimited fines and a maximum of two years' imprisonment for offences involving copyright infringement. Software companies are increasingly keen to see the Act enforced. For instance, lawyers acting on behalf of FAST recently applied for a High Court order which allowed an unannounced inspection of a newspaper's premises and resulted in an out-of-court settlement believed to be £30,000. Since 1988, FAST has seized pirated software worth over £1M, and is currently investigating nearly twenty organisations in the UK, ranging from banks to local government offices.

At present, there are widely different attitudes to the protection of software in countries within the EEC and not all countries have updated their legislation. For example, copyrights will only be granted in Germany where the software has a significant degree of originality. (This may explain why there has been greater copying of software in Germany than in the UK). Because of these different approaches, the EC issued a software Directive (91/250/EEC) in May 1991 designed to harmonise legal protection of software throughout the Community; the Directive must be implemented by Member States by 1 January 1993. The 'grey' areas of interfaces and decompilation engendered much heated debate within the software industry and the Directive represents a compromise between the disparate views. Debate continues on interpretation of the Directive's text and related issues are now being raised in the national Parliaments of Member States.

CURRENT ISSUES

Software and Copyright

There is debate over the **nature of software itself** and the appropriateness of using copyright law for its protection. Copyright is intended to protect only the *expression* of ideas - not the ideas themselves. Thus in the field of literature, it is not illegal to develop another author's ideas and the world's literary traditions owe much to the heritage of past writers. However, it may be an infringement to copy the plot of a novel, even though the characters might be changed.

How does software fit into this legal framework? Software is treated in the same way as literary works which are protected in whatever language they are published; thus both the source and object codes for a program (see Box) are deemed copyright material. Software companies guard their source codes as trade secrets and many of them regard decompilation of object codes by another party as an infringement of copyright. They see allowing decompilation as potentially enabling a competitor to find out the logical structure of a program and develop an equivalent cheaper product. Organisations like FAST and SAGE point to the high development costs of original software as justifying a very high level of legal protection for their members' products. They see any loss of protection as reducing the rewards for innovation and incentives to develop novel software.

Other software specialists claim that the practice of decompilation is as old as computer programming itself and that it has not had a deleterious effect on a thriving worldwide software industry. Many UK and other European software companies offer products which require interfaces with the most widely-used software products and computers, which are dominated by large companies abroad (particularly in the USA). Their concern is that too stringent a restraint on decompilation will lead to the development of the smaller European software industry being obstructed, to the advantage of overseas competitors.

There is also a fundamental difference between software and other copyright material in that software has a *functional* role (controlling computer operation), in contrast to books, plays, etc. (even a recipe book has only an instructional value). It has thus been suggested that software be treated as a special case under intellectual property law (*'sui generis'*). One precedent would be the special status afforded to the design and layout of Integrated Circuits under a Treaty drawn up within the World Intellectual Property Organisation (WIPO).

Many however, prefer to allow existing legal procedures to cope with advances in technology rather than define new categories of protection. Common copyright standards are afforded in more than eighty nations via the Berne Convention and many countries have also signed a similar Universal Copyright Convention. Under the existing regime therefore, although there may be national variations, a program has widespread international copyright protection from the moment of its creation. In contrast, it might take many years to establish a new international regime specifically for software protection.

1. This is well-established in the personal computer market where 'clones' of machines produced by leading manufacturers are made by others. To do this, engineers probe the detailed working of a particular machine and then hand on a precise specification to a totally separate engineering team. The latter then produce their own version - a clone - of the original hardware. Elaborate legal safeguards are taken to ensure that the cloning engineers cannot be accused of plagiarism. Precisely the same principle can be used to create a 'cloned' piece of software.

Software development and Interoperability

The large market based on developing or extending applications for major software products has already been mentioned. Indeed, many of the larger computer software and hardware companies have not objected in the past to other companies making 'add-on' software, since such accessories can broaden the appeal of the original products and increase sales. However, a different view may be taken when a software company wishes to make a *competing* product. The task of making connecting or competing software products almost inevitably involves decompilation of the original program, and it is over the extent of this that the main debate centres.

Some argue that the freedom to decompile parts of another company's software is desirable. In this view, decompilation should be distinguished from copying since, once the interfaces have been deduced, the programmer writes his own software to perform the required functions quite independently of the original source, and may display as much ingenuity as the original programmer. By allowing the creation of competitive products, it is argued that decompilation ultimately lowers costs to the user¹.

Major software companies normally publish details of the more important interfaces used, and argue that it is therefore unnecessary for any other company to decompile their programs. ECIS argues however that published details may be inadequate, or may omit applications unforeseen by the original author. They thus consider that decompilation for the purpose of identifying interfaces should be allowed in law, even to the point of replicating lines of code from the original in the 'add-on' or substitute product. The compromise reached in the EC Directive allows interface characteristics to be deduced by decompilation, but only as a last resort (i.e. where details cannot be obtained by approaching the original manufacturer) and not for the purpose of creating a substantially similar program.

There may thus be scope for differing interpretations of the circumstances under which decompilation can be carried out; for instance, where a company might stand legally if it has decompiled parts of another program because the technical guidance provided on interfaces is insufficient or is delayed by the other party.

In the USA, producers of competitive software have been challenged increasingly in the courts over alleged infringement of copyright. Some lawsuits have been based on what is termed as a program's 'look and feel' - the way in which information is displayed on the computer screen and which keys on the keyboard are used for certain operations. Some US courts have ruled that copying 'look and feel' characteristics constitutes a breach of copyright. This has serious implications for

makers of competing software, since an objective of the programmer may well be to ensure that his product does not require the user to employ new and unfamiliar techniques; there is thus a tendency to adopt similar 'look and feel' characteristics to programs already in widespread use.

Court challenge can have severe consequences on a software company both through legal costs and in delaying the launch of a product. Even the threat of litigation can effectively 'destroy' a piece of software without the issues ever being decided at law. The market opportunity for software can be very short and distributors will not carry a product which may be the subject of a court case. Some see this trend of court challenge over 'look and feel' as threatening to reduce competition in the software industry. They see the questions raised as not dissimilar to those over the layout of a typewriter and computer keyboard where the familiar QWERTY arrangement can be legally copied - otherwise different typewriter manufacturers would have to lay the keys out in a different order. Others, however, point out that the commercial success of a program is often a result of its ease of use (i.e. look and feel) and vigorously defend every related aspect.

Many fear that court challenges will increase in the EC after Member States' implementation of the Directive by 1993, both by European companies and between US software houses and EEC software firms and users. However, others point out that US² law is different from that implied in the EC Directive, and that comparisons with look and feel cases in the USA are unnecessarily alarmist in the European context. If legal action does take place, the first priorities for software companies may be to act in countries where legislation has been weaker than in the UK.

Changes in UK Practices and Legislation

Software audits. So that computer-users in the workplace cannot be accused of pirating software, it will be increasingly common for user companies to carry out 'software audits', i.e. logging all items of software, their licences, and the computers on which they are mounted. FAST is working to bring to the attention of managers and employees the implications of the CDP Act and to provide help with methods of software auditing. Despite the ease with which software can be copied (and erased), FAST's experience suggests that the CDP Act can be successfully policed.

Legislation. The DTI believes that relatively minor revision of the CDP Act will be required as a result of the EC Directive. Regarding decompilation, current law allows copyright material to be reproduced for the

purposes of research or private study. This so-called 'fair dealing' concept is believed to offer the flexibility to allow decompilation for the limited purposes described earlier. ECIS however believe that the law should be amended to make explicit mention of decompilation as a legitimate activity under the fair dealing principle and this approach has not been ruled out by the DTI.

A further principle in copyright applies where there are only one or two ways of expressing an idea (for example, there is only one way to structure a traditional balance sheet). In such cases, the original 'idea' and its 'expression' have merged and copyright protection is not appropriate, since it is against the spirit of copyright law to protect an idea. There is no provision for this 'merger doctrine' in current UK law, and ECIS argue that the CDP Act should be amended to include it. Without it, they see increased scope for legal challenges of the type already going on in the USA. Others see the incorporation of the merger doctrine in law as unnecessary in view of the fair dealing principle and believe that it will add further complications to an already complex part of copyright law.

Further points at issue concern how far new regulations should seek to provide a definition for terms which have not been defined in the Directive. For instance, to some, 'interoperable software' means add-on products only; to others it includes software which can be a substitute for the original software. Since the EC provided definitions for only a limited number of terms, there is considerable scope for disputes on interpretation to arise if the key terms are not defined in legislation. At present, the regulations stemming from the Directive on software are being drafted by DTI which intends to circulate draft regulations for consultation in the first half of 1992.

Patents. There are also increasing attempts in the USA, some of them successful, to apply for patents on computer software. In contrast, the European Patent Office has recently restated that patent protection can rarely be applied to software. This is because computer programs rarely possess sufficient 'novelty' to fulfil the requirements for patent action (though exceptions exist where the program is part of a system or machine that itself performs some innovative operation or process).

FURTHER READING

Additional details and background information are available from POST, 2 Little Smith St., London SW1P 3DL, tel: (071)-222-2688.

2. Legal judgements in US courts have not been consistent, which may reflect the technical complexity of software itself and the difficulty of applying legal provisions to the 'grey' areas of software development.