# Kaltons' Response To The European Commission Consultation Invitation On The Patentability Of Software – December 2000

## 1. Our Thoughts: Economic & Commercial

This is the collective view of Kaltons IT/Internet department.

We have for many years been aware of the argument against allowing software to be patentable. In particular, we refer to the commonly subscribed view that if software were patentable, it would enable large corporations to inhibit innovation, particularly in the context of the Internet. The argument largely consists of the assumption that those companies would hold the sector to ransom.

On one hand, as an Internet specialist firm we naturally share the concern that nothing should be allowed to impede the development of technology and especially the Internet. The possibility exists that in limited circumstances companies may suppress advancement of technology but we believe that these fears are exaggerated. Moreover, we also question the proposition that if software becomes generally patentable there would be an administrative crisis with hugely-increased software patent applications. The US, is able to cope, therefore, so should Europe. However, it is conceded that the real difficulty examiners would face is in respect of dealing with technical issues such as "novelty" and "inventive step".

On the other hand, we have the far greater fear that the economic interests of the European Union are more likely to be adversely affected by the failure to encourage investment in technology within the EU. We consider the potential negative aspects of allowing software patent to be registered are totally outweighed by the potential benefits, namely in attracting forward-looking, innovative companies to the EU. Given a choice between Europe and another jurisdiction permitting software patents, why would anyone choose somewhere where the reward for their research and development would be exceedingly limited?

Furthermore, patents are hard to acquire and expensive to maintain and of relatively short duration. In other areas of innovation they have encouraged substantial investment in research and development and have, by dint of the law governing patents, not generally stifled innovation. We see no reason why the position should be any different for software and, in particular, for businesses involved in the development of the Internet.

## 2. Our Thoughts: "Technical"

The main "technical" problem with the present European patent regime is that it (as a general principle)excludes software from patentability. Although over the last 15 years the Technical Board of Appeal of the European Patent Office (EPO) has tried to get over this hurdle, almost to the point of, in effect, overturning the clear intent of the regime, it is still shackled by its current provisions. In contrast, TRIPS (Trade-Related Aspects of Intellectual Property Rights Agreement 1994), to which the European Patent Office is not a signatory, has abandoned the exclusion of computer programs from patentability. For all the "commercial" reasons we have so far mentioned, we take the view that the current European regime must adopt the TRIPS approach as to make it possible to patent software provided it meets the very basic requirement of patentability, which in our view should not be disturbed in any way, shape or form.

If the non-exclusionary approach in TRIPS is followed, it would be possible for the European patent regime to deal with both the *symbolic* and *functional* aspects of software, without being snared by the exceptions which we have to deal with today like the "as such" exception.

Moreover, the fact is that at present the EPO grants many thousands of computer programs a patent whilst maintaining the fiction that software is not patentable, and the TRIPS approach will put a stop to that dilemma.

In order to show how we arrive at this conclusion we need to in the foregoing set out the background of the European patent regime and its development until now.

### 3. Patent Protection of Software: The Background

In the development of intellectual property law, aesthetic creations ( e.g., literary and artistic works) have traditionally been protected by the law of copyright on the basis that copyright's main purpose is to encourage and reward original and creative works. While the protection of copyright is extensive and welcome, it does not protect the owners of software completely. For example, it is easy to ascertain the latent ideas and "copy" these without literal copying of any of the code used in the original. In this context, patent protection would clearly be welcome. This is because patent grants its holder monopoly protection against reproduction of the novel ideas that copyright simply does not protect.

However, industrial property has attracted protection such as designs, patents and trademarks on the basis that protection of investment in useful new technology would provide an incentive to finance research and development.

The critical point of departure which has resulted in the muddied waters of the current European software patent law was the understandable failure in the 1970's to appreciate that software has an industrial aspect to it aside from its obvious aesthetic dimension,

### 4. Is Software "Aesthetic Creation" or "Industrial Property"?

There are, however, two aspects to software. It is both *symbolic* (and therefore aesthetic in nature) as well as *functional* (and therefore 'industrial'). Accordingly, in respect of the symbolic aspect of programs, they should surely be protected by copyright as a type of aesthetic creation such as a literary work. However, in their functional aspect (i.e., the way that they behave) it appears more appropriate to protect them as types of industrial property. It is the failure of the European regime to underscore this dual aspect of software that has resulted in software being excluded from patentability. If this had been historically better appreciated, then in respect of the *functional* aspect of software, we believe patentability would have been extended.

### 5. The Traditional Attitude of European Patent Law *vis-à-vis* Software

Traditionally, software has been protected by copyright and excluded from patent protection in Europe as can be seen from the exclusion of aesthetic creations from patentability in article 52(2) of the European Patent Convention 1973 and as incorporated under section 1(2) of the Patent Act 1977. This was mainly because, from a lawyer's perspective, the human-readable source code appeared symbolic as opposed to functional, and because source code listing looked like literary work, copyright protection was naturally extended to it.

However, in so far as the machine-readable object code was concerned, although it posed difficulty being functional in nature, copyright lawyers dealt with it by treating it as a translation from one language (human-readable) to another (machine-readable) and extended copyright protection accordingly.

European Patent Convention 1973 ("the Convention"), excluded software from patent protection on the basis that all it consisted of was abstract and intellectual mechanisms as opposed to useful tangible products or processes. Patentability requires a specific technical application, but software is not technical in nature.

The critical failure of the Convention was in not going beyond, as a general principle, the non-technical nature of software *as such*. If it had gone further to focus on the technical effect that software gives rise to, that aspect would have readily attracted patent protection. However, as an exception, as will be seen below, the Convention did recognise this.

**6. The "as such" Exception**

Article 52(3) of the Convention provided an exception or loophole. It states that that the exclusion only prevented anything from being treated as an invention to the extent that a patent or a patent application relates to software *as such*. In other words, the exclusion from patentability applied only to software "as such". Therefore, if a software patent application is made for something different from a software program as such, the application would be treated as a genuine patent application and not *prima facie* excluded. The *as such* exception shows that the exclusion of software from patentability is in respect of form rather than substance. So, it is not permissible, without more, to seek protection for a computer program when it is stored on a magnetic medium or when merely loaded into a computer. However, it is permissible to seek patent protection in respect of what the computer – into which the program is loaded and which controls it – is doing!

So, for example, in *Merrill Lynch Inc's Application* [1988] RPC 1, there was an application to patent an improved data processing system for arranging, analysing and dealing in customers' stocks and implementing an automated trading market for one or more securities. The program could be used in conjunction with any appropriate computer to process dates relating to transactions in stocks and bonds. Falconer J in the English High Court held that an invention must be found in some aspect of the application aside from the computer program itself. Although this decision was subsequently criticised by the Court of Appeal in this and other cases, the distinction drawn by Falconer J between the computer program *as such* (which is excluded from patentability) and the way it is applied (which should be allowed patentability in the normal way) is illustrative of how the "as such" exception should work.

So some aspects of programs become patentable. This is because the Convention allows claims to novel methods of programming computers to operate in a particular way because these are not inventions in a program *as such*. Accordingly, if a claim is expressed in terms of improved and modified apparatus operating in a new way, it may be patented The task performed by the computer program becomes the part which could be patented but not the contribution made by computer program itself. So, for example, if a claim is expressed in terms of equipment operating in accordance with a program's instructions, it may become patentable, but not the algorithmic steps instructed by the implementing programs.

In *Vicom System's Application* (1987) 2 EPOR 74, the Technical Board of Appeal of the European Patent Office, held that since the relevant application to patent a 'program' was for a technical process rather than a program, it was patentable. It was further held that the fact that the inventive aspect of the claim resided in the program was no bar to patentability.

Subsequently, the European Patent Office (EPO) itself tried to clarify the meaning of this new concept in its official guidelines. Once it gave the following examples: "[P]rogram controlled machines and program controlled manufacturing and control process should normally be regarded as a patentable subject matter." (*European Patent Office Guidelines*, C-IV, 2.3 xii.1994). In other words, where programs have application, such as an industrial machine or process controlled by a program, then the overall process considered as a whole could be patentable.

In the subsequent case of *Koch & Sterzel* (1988) EPOR 72 it was held that the invention must be assessed as a whole. If it makes use of both technical and non-technical means, the use of non-technical means (i.e., the internal effect that a program has on a computer) does not detract from the technical character of the overall claim.

## 7. Is a program that only has an internal effect on a computer patentable?

The answer given by EPO via its guideline was that: 'Where the claimed subject matter is concerned only with the program controlled internal working of a known computer, the subject matter could be patentable if it produced a technical effect" (European Patent Office Guidelines, C-IV, 2.3 xii.1994.)

In the *IBM Cases* [1999] RPC 861 [Case T1173/97 and Case T0935/97], the Technical Board of Appeal of the EPO held, however, that simple changes in the physical state of hardware, for example, by the manipulation of electrical currents, are insufficient. In order to be patentable, what is required is something more than that. There must be a further technical effect. The Board of Appeal observed: [*IBM's Application* [1999] *RPC* 861 at 871]:

> '[A] patent may be granted not only in the case of an invention where a piece of software manages, by means of a computer, an industrial process or the working of a piece of machinery, but in every case where a program for a computer is the only means, or one of the necessary means, of obtaining a technical effect within the meaning specified above, where, for instance, a technical effect of that kind is achieved by the internal function of a computer itself under the influence of said program.'

It has been argued that it 'is astonishing that the explicit provisions of the Convention denying protection to computer programs have been read into virtual non-existence' [See Tapper in *Masons Computer Law Reports* 280]. Thus, whereas the position under the Convention was that a computer program *as such*, was not patentable unless it could be brought under the *as such* exception in article 52(3) of the Convention and section 1(2) of the Patent Act 1977, now, however, the position is reversed. The position today appears to be that a program is patentable unless it can be shown that the patent application is for a computer program *as such*.

This is clearly unsatisfactory and amounts to judicial legislation and activism of a kind that may even cause some US courts to blush. The better way of going about making software patentable is to legislate to remove the general exclusion of software from patentability under article 52(2) of the Convention in line with the TRIPS approach.

## 8. Trade-Related Aspects of Intellectual Property Rights Agreement 1994 (TRIPS),

Article 10(1) of TRIPS states that computer programs, whether in source or object code, shall be protected as literary works. However, Article 27(1) provides that:

> 'Subject to the provisions of paragraphs 2 and 3, patents shall be available for any inventions, whether products or processes, in all fields of technology, provided that they are new, involve an inventive step and are capable of industrial application.'

The exclusionary provisos under paragraphs 2 and 3 do *not* exclude computer programs but other things like immoral inventions, medical treatment etc. The clear words of Article 27(1) suggest that computer programs have the same ability to be patented as any other product or process.

Unfortunately the EPO is not a signatory to TRIPS and therefore it cannot be made to adopt the concept accepted by TRIPS that computer programs are not excluded from patentability as a general rule, as it is bound by the Convention. However, the EPO is trying to catch up with TRIPS as judiciously as possible. In one of the IBM cases it was said that:

> '[A]lthough TRIPS may not be applied directly to the EPC, the Board thinks it appropriate to take it into consideration, since it is aimed at setting common standards and principles concerning the availability, scope and use of trade-related intellectual property rights, and

therefore of patents rights. Thus TRIPS gives a clear indication of current trends.' IBM's Application [1999] *RPC* 861 at 868.

The EU *Commission's Green Paper on the Community patent and the patent system in Europe* (1997) 16-17 has recommended that TRIPS be followed.

**9. Conclusion**

- There are very strong economic and commercial reasons why as a general rule software should be made patentable in Europe. Principally, we believe the impact on the attractiveness of the EU as a base for technology-based businesses as opposed to currently more "friendly" jurisdictions such as the US is the most compelling commercial reason for giving serious consideration to the removal of the general exclusion of software from patentability.

- Even though it clear why it is that the current legal regime in Europe has ended up making software as a general rule unpatentable, there are sound reasons and clear trends to warrant that software should be made generally patentable. The technical stumbling blocks inherited from the 1970s, when software was viewed quite differently, need to be overcome by legislation.

- The EPO and the Technical Board of Appeal of the EPO have bent as far backward as they possibly can to arrive at a position where software is generally patentable. However, this is unsatisfactory because they have achieved this by apparently flying in the face of express legislative intent.

- It is clear that the model that Europe should follow should be the TRIPS model which does not, as a general proposition, exclude software from patentability as does the Convention. TRIPS puts software in the same basket as any other invention, whether product or process, and so long as the basic elements of patentability are met – viz. novelty, inventive step and industrial application – a computer program should be patented as all other products in other areas of innovation. That is what we wish to see and that is what we ultimately recommend in this submission.

*[This submission had been prepared by senior IP consultant, Louis Joseph*[1] *and the other members of Kaltons' IT/Internet]*

[1] MA, LLB, of Gray's Inn, Barrister, Advocate & Solicitor of the High Court of Malaya and the Supreme Court of Singapore, ~~Legal Counsel, Horecast Holdings Limited (BVI), Singapore [mskj1@singnet.com.sg]~~Senior IT/IP Consultant at Kaltons, Solicitors, London – www.kaltons.co.uk - [louis.joseph@kaltons.co.uk]