

YOUR SOFTWARE AND HOW TO PROTECT IT

a guide for small businesses on how to
protect the software you have developed



Contents

Will this document be useful to you?	1
Who develops software?	1
Protecting your software	2
Do you need to protect your software?	2
Informal protection	3
Formal protection	4
Copyright	4
Trade marks	6
End-user Licences	7
Patents	9
Annex 1: Registering a trade mark	12
Annex 2: Patent searching	12
Annex 3: How to find out more about patenting	13

The guide is based on research by experts on Intellectual Property Rights at the University of Sussex and the University of Sheffield. The research was funded by Directorate General Enterprise of the European Commission.

Standard Disclaimer:

The views expressed in this brochure are purely those of the writers and may not in any circumstances be regarded as stating an official position of the European Commission. Please note that the information provided in this document is to be used only as a guide.

Will this document be useful to you?

This document will be useful to you if you develop software and if you need to find out if you should be protecting it. If the answer to both of these questions is yes, read on: you will be guided through the various formal and informal types of protection available to guard the intellectual property invested in your software, and the benefits, costs and risks attached to each.

Increasing numbers of small businesses in Europe are developing software, either as their main business or for their own IT systems. Many are unsure how to protect the intellectual property held within the software they have developed.

Protecting software is one of the crucial steps in making profits from it or preventing others from doing so. This document provides a clear guide to the four types of formal legal protection provided by copyright, trade marks, licenses and patents. It also briefly outlines the main forms of informal protection used to protect software.

These forms of protection can complement each other. For example, patented software can also be given further protection by a trade mark. However, each form of protection brings costs, so this document highlights these costs and provides the information that you need in order to get enough protection without incurring unreasonable cost.

The development of software is a major business sector. Many firms specialize in developing software products for business and home use. But many other small firms go well beyond simply buying their IT systems 'off the shelf'. Many find that they develop their own software. Sometimes this is to maximize the productivity of their existing IT systems, sometimes they cannot find software to fulfil a particular function, and sometimes they have the specific aim of generating revenue by selling their software to others.

So two types of firm develop and sell software: those for whom it is their main business, and those that have developed software for their own use and later realize that it has commercial value. Whichever of these you are, the central question is: Do you need to protect it?

protecting software is crucial in order to profit from it

Who develops software?

Protecting your software

Do you need to protect your software?

establishing copyright can be useful in signalling your expertise

Two basic approaches to protecting your software

many small firms use both formal and informal means to protect their software

There are three reasons why you might want to protect your software:

- **to make a financial return from it**
- **to prevent others from making a financial return from it, or**
- **to prevent others from using it.**

You may be happy to allow others free access to your software. You may not see a commercial application for it.

Alternatively, you might want to allow other software developers to help improve it. The latter is often achieved through the Internet - so-called 'open source' software. If this is the case, you may not need to protect your software, although you are still entitled to copyright in those parts of the software that you have developed. This can be useful in signalling your expertise to others, even if you do not want to derive any direct financial benefit.

You will need to protect your software if you want to make a financial return from it. The rest of this document is devoted to helping you if you are in this situation.

If you've read this far you have probably decided that you do need to protect your software. This document now reviews the two main types of protection available: formal and informal. Formal means are those recognized by the law and include copyright, trade marks, licences and patents. There are also a host of informal means, however, including first mover advantage, technical protection, trust and market knowledge.

Our research has shown that many small firms make use of both of these types of protection for their software. Each of these types of protection has benefits, costs and risks. The aim of this document is to set out these benefits and costs in the light of people's experiences with using them. In protecting your software, there is no point in incurring expense unnecessarily. We also aim, therefore, to encourage you to decide which method fits your needs best.

The document dwells mostly on the formal legal means of protection, since it is these that generate the most confusion among business decision-makers. However, we start by outlining some of the informal means of protecting software that are commonly used.

Informal protection

Research among small firms shows that many see an important role for informal methods in protecting their software. Many are aware of the costs of pursuing legal redress for infringement of, for example, copyright. In this context, firms are inclined to use a mixture of both formal and informal methods to protect their software. This section briefly outlines the main types of informal protection.

Firms often use 'technical' systems for protecting their software. These include: encryption; dongles (adjustments to computers to prevent infringement); steganographic techniques (the most common being to insert extra pixels into the spacing of a document which can be picked up by the originator with special software); firewalls; and passwords. However, managers in small firms acknowledge that such systems can be complicated and user unfriendly.

There are also various forms of non-technical means for protecting software. These include seednames, which are fabricated names that appear on copyrighted mailing lists. These entries have addresses to which the list's original owner has access. This means that if the mailing list is used unlawfully, the originator will know. Non-technical protection methods also include a reliance on collecting societies and trading standards officers to detect software pirates. Collecting societies are professional copyright organisations that register and licence copyrighted material.

Firms often cite the fact that they are in a small market niche as a form of protection. Similar to this is the use of trust as a form of protection. Relationships grow up between producers, users and other producers, thereby increasing the threat of bad publicity for potential infringers.

Finally, a similar market-based technique is first mover advantage. Software is a fast moving field, so many firms see their best defence to be to develop and market new versions of their software rapidly. Given the knowledge needed to make software products, pirates will be left behind as they struggle to catch up and offer software with competing levels of functionality.

many small firms see an important role for informal methods in protecting their software

informal methods include first mover advantage, technical protection, trust, and market knowledge

Formal protection: copyright

Introducing formal protection

There are four main formal legally established ways to protect software. These are, in increasing order of legal strength but also in terms of cost and complication: **copyrights, trade marks, licences and patents.**

Each of these forms of legal protection are now reviewed. You will find out what they are, how you get them, how they work and the type and strength of the protection they provide as well as their costs and any weaknesses.

As will be explained, copyright is automatic, so it can easily be used: all you need to do is make clear your claims to copyright. Trade marks are relatively inexpensive and are widely seen as a highly effective form of software protection. They build on and complement copyright. Licences and patents can be expensive and complicated, but they similarly can work side by side with copyright and trade marks.

Copyright

Copyright is the legal protection automatically given to original literary, dramatic, musical or artistic works; sound recordings, films, and broadcasts; databases; and computer programs.

*almost everywhere,
the act of writing a piece of
software automatically
gives you copyright*

Here's the good news: almost everywhere in the world, the act of recording a piece of software automatically gives you copyright as the writer of it. Recording may take any form be it in writing or simply storing it electronically either on disk or incorporated into hardware. This right applies to citizens of all of the nations that are members of the Berne Copyright Union, and this includes most countries including the EU, the US, Japan and many others.

Copyright is therefore automatic, cheap to display and almost universally recognized.

It is advisable to include prominent copyright notices on the versions of the software you sell to end users. These should appear on the wrapping, the disks, and on the first page to appear on the screen. It should say '© [name] [year] all rights reserved'. Not putting © [name] [year] will not diminish your legal protection, but it is still helpful to display your claim.

How does copyright work for computer software?

Copyright provides you with rights as the person who created the software. It basically means that if anyone uses your material without your permission, you have the right to sue them for financial recompense. In the same way, copyright also protects the graphics and original layouts you develop for the program. Although copyright can protect the original graphics and layouts you develop for your program, it may be possible to acquire further protection in some jurisdictions by registering them as designs. This has the advantage of providing clear evidence that you are the owner of the rights in these elements of the program.

Suing for copyright infringement can be quicker and relatively cheaper than for patents. Most cases start and finish with an injunction or similar provisional measure to stop unauthorized use of your material. Such injunctions can usually be obtained rapidly in European Union countries. However, litigation and court proceedings can sometimes be long-winded and therefore expensive.

Formal protection: copyright

It is important to establish ownership of copyright. Although software created by employees belongs to the employer, software created by outside consultants belongs to them even though you have paid for it, in the absence of any evidence to the contrary. This also applies to graphic designs. Consequently, you must ensure that there is a written letter or contract signed by or on behalf of the consultants clearly stating that ownership of the copyright is assigned to you. The requirements or formalities for such a letter or contract vary in different countries, so you should consult a lawyer or an attorney if in doubt. Failure to secure such a commitment might mean that you might find you have no authority to use the software as you intended. In such circumstances without a proper licence, you might be considered an infringer.

Keep evidence of who created the various parts of the software. Date these notes and keep them, along with the source codes, in a safe place. In some countries this can be done by depositing the source codes with your bank. If this option is not available, or if you prefer it, a reliable third party such as a lawyer can be used. Whatever way you choose, make sure that the third party provides evidence of the date of receipt so that the state of the source codes at the relevant date is certified.

Copyright protection is free, but its weakness is that it only protects you against having your software copied, translated, adapted, rearranged or otherwise altered and distributed. You cannot sue for copyright infringement if another developer comes up with the same solution independently. This is quite possible given modern standard programming procedures.

Copyright also does not prevent someone else developing and marketing a competing package by using your novel idea, as long as they do not directly copy (or do any of the other acts listed above) any aspects of your software. They can do this so long as their product does the same job as yours without copying the source codes or object codes of your software. If you have spent a large amount of money developing a package, and there is a danger of this happening, you need to look at patent protection.

Things to watch out for: copyright

ownership of copyright must be established in writing when you use outside consultants to develop software

Formal protection: trade marks

Trade marks

trade marks are relatively cheap and widely recognised

The second form of legal protection is a trade mark. These normally consist of a distinctive product name and a logo under which you are selling the program. Similar to copyright, trade marks are both relatively cheap and widely recognized. For some well-known software programs, this form of protection has proved to be the most useful in practice.

A good software program gets known by its name. In most countries, you can simply declare that this name is your product's trade mark. However, although trade marks, or other signs used in the course of trade, are protected in most countries without registration, registered trade marks are much easier and cheaper to enforce.

Registering a trade mark is usually straightforward and inexpensive. How you go about registering them is dealt with in Annex I.

How does a trade mark work?

for some well-known software programs, trade marks have proved to be the best form of protection

If you have registered your product's name and any distinctive representation of it as a trade mark, rivals are not able to use any name or representation so similar that customers might buy their product, mistaking it for yours.

If you have not registered the trade mark, you can still pursue pirates who try to "pass off" their trade mark as yours by altering minor aspects of your trade mark that may not be apparent to the customer. This process is, however, more complicated and expensive to enforce than that of a trade mark.

Infringement of a trade mark is usually easier to detect than infringement of copyright, especially in the case of software. This is because trade mark infringement is immediately visible to the eye, whereas copyright infringement will largely have to be established by comparing each part of the software for unauthorised copying.

Things to watch out for: trade marks

If you have used an agency to design your trade mark, you need to take the same steps outlined under copyright to assert ownership. That is, you have to ensure that you have a written letter or contract signed by or on behalf of the consultants clearly stating that ownership of the trade mark and copyright is assigned to you.

The weaknesses of trade marks are similar to those for copyright, in that they do not prevent others from copying the core idea and providing a similar end product through a different means.

Formal protection: licences

You can require users of your software to sign an 'end-user licence' before they use it. In a licence, you can set the terms under which people use your software, including what uses are permissible. Since you can require users to agree to the terms before they can gain access to your software, this can provide a valuable form of protection for your software.

Licences have legal recognition and protection to varying degrees in different countries. Unlike copyright, trade marks and patents, however, there is no international agreement covering the design and status of end-user licences. It is important to bear in mind that end user licences are not a form of intellectual property right in the way that patents, copyright and trade marks are regarded as part of Intellectual Property Law.

Licences can cover a wide range of issues. Many of these can be confirmed by looking at the licence agreement that comes with any widely-used software package.

First, and most importantly in terms of the protection provided, the licence should make it clear that all intellectual property embodied in the software belongs to you, including copyright, patents and trade marks. In addition, if you are allowing access to the source codes for any purpose, the user must be required to sign a confidentiality agreement in advance.

You can also provide for termination of the agreement in the event that the user is in breach of any term of the licence and in certain other cases. Some terminating events should be automatic e.g. bankruptcy of the end-user. This is because a third party that may take over the user's assets will not necessarily be bound by the terms of the licence. Some events may entitle you to serve notice, so that the licence will terminate if the licensee has not complied within a certain period.

The licence will also need to address the number of copies that can be made: obviously distinguishing between the use of the software on an organisation's network and by individual users. Where special rates are given for educational and non-commercial uses, the licence should prohibit commercial uses.

Under the European Commission Software Copyright Directive, you cannot restrict by contract the following:

- the making of back-up copies necessary for use of the software;
- use of the software to observe, study or test the functioning of the program in order to determine the ideas or principles underlying it; and
- in certain limited circumstances, the decompilation (reverse engineering), i.e., copying of the code and translation of its form to obtain information necessary to achieve interoperability of an independently created program with other programs.

For a full description of the restricted acts, you may want to refer to the Directive in the **Official Journal of the European Communities** No. L 122, 17/05/91 p. 42, or via <http://www.ipr-helpdesk.org> which is an initiative funded by the European Commission.

End-user licences

What can a licence cover?

licences allow you to set the terms under which people gain access to your software

under European law, licences cannot restrict certain significant activities

Formal protection: licences

For networked software, the following issues will need to be addressed:

- a licence fee payable at regular intervals e.g. monthly.
- payment for your personnel providing expert advice. If paid for separately, provision for the payment of value added tax or its equivalent will be required.
- payment for upgrades, where relevant.

A licence can also have defensive uses for you, in excluding liability for loss or damage arising from use of the software. The effect of such disclaimers may be limited in some jurisdictions. You should discuss this issue with product liability insurers.

Finally, the licence can also provide for receipt of your bulletin dealing with upgrades, errors and 'patches'.

Other licensed uses

Where software has a number of applications, but you yourself are interested only in licensing a particular one in the field where your own expertise lies, you can sometimes find other income streams by licensing others to develop alternative applications of it. In this case the most important thing is a confidentiality agreement relating to the source codes because the licensee will require access to these in order to develop other applications. In most jurisdictions, this can take the form of a simple letter signed by the licensee. Again, it is advisable to seek local legal advice. You should also address the issues set out above on end-user licences.

Things to watch out for: licences

Given that it is quite straightforward to design and implement software end user licences, software should never be sold without requiring the end-user to agree what uses are permissible.

The problem is that the terms of the licence are dependent on the law of the jurisdiction in which the software is sold. There may, for example, be requirements that the contract is in the local language, or it will not be binding on the end-user. This is not too difficult in the case of software being sold over the counter on disk. You will need a local lawyer to draft the terms in the local language, and ensure that when the program is installed the licensee agrees to the terms, which appear on the screen in the normal way.

if selling your software in other countries, licences may meet barriers such as a local language requirement

In the case of software delivered on-line, the problem is much more difficult to solve. The problem is that the terms of the licence are dependent on the law of the jurisdiction in which the software is sold. Another problem is that some jurisdictions require a signature and do not recognize electronic signatures. At the very least, you should consult a lawyer in each of the principal jurisdictions/markets in which you wish to make the software available on-line. Inserting a requirement that the software be governed by your own law is only of limited assistance if you cannot pursue infringers in their own jurisdiction.

Most importantly, you should consider technological protection measures, i.e., anti-copying devices within the software, which would guard against illegal copying.

Formal protection: patents

A patent is a legally enforceable statement of ownership of an invention. Patents can protect any patentable invention that often gives market value to your program. A granted patent gives you an exclusive right (ownership) to your invention, in exchange for a complete description and publication. In this way, the exclusive right granted by the patent prevents others from using your invention without your approval. Such inventions can involve either products or processes.

Patents do have one other significant advantage: there is evidence that they make it easier to obtain venture capital and commercial partners. Even a filed patent application in a single country or patent jurisdiction may help in this regard, and such a filing is quite cheap. But remember, patent protection is territorial in nature: for instance, if you file in the UK, you will only receive protection in the UK.

In applying for a patent, you will need to do the following:

1. **check** if your invention has not yet been published,
2. **find out** whether your invention will qualify for a patent by using the services of a patent attorney, and
3. **employ** a patent attorney to conduct the specialist work involved in the patent application.

To ensure that your idea has not already been published, you should undertake a search in various databases. Patent databases offer a vast amount of information and must systematically be consulted. Advice on using these databases is given in Annex 2. Many people find that their own knowledge of the industry and its products is at least as important as searching patent databases, because much software is not patented. A search of the relevant academic scientific literature can also be useful.

A patent search can also give you some indication of similar or identical inventions that have already been patented. This search can enable you to identify patents still in force whose claimed product or process may cover your invention. In such a case you should rely on professional help to interpret the results before proceeding further. It is important to note that at present, US patents are only published when granted, and in consequence there are rather a lot of unpublished patents waiting to surface. However, this situation is bound to change because a recent amendment to the US Patent Act might eventually result in having more than 80% of US patent applications being published after 18 months from time of application, like in Europe and most other countries. However, if you find that your idea has already been patented but you were in fact the first to invent, under US law the first to invent is entitled to the patent, so you can fight out an 'interference suit' to prove this. This can, however, be an expensive process!

Patent databases can be valuable sources of information in developing new products, but beware because developing new products from patent information can infringe the rights conferred by the patents. You should always seek advice from a patent agent about what you can do legitimately.

Patents

a granted patent gives you an exclusive right to your own invention

How do you get a patent?

Has your invention already been published?

many find that their own knowledge of the industry is at least as important as searching patent databases - much software is not patented

Formal protection: patents

Does your invention qualify for a patent?

whatever your software invention is, it is useful to consider patent as a protection tool.

In order to qualify for a patent, your idea must be novel. In most patent offices of the world this means it must be absolutely new in world-wide terms. It must in addition represent a significant step forward on the prior art: it must involve an 'inventive step'. The results of the prior art search described in the previous paragraph should ensure that these conditions are met.

Some patent offices exclude software from patentability. The present position is that three of the world's major patent offices, those of Europe, the US and Japan, do allow patenting of some software, although there are differences between the criteria applied in accepting applications in these offices. For example, all novel and non-obvious software which produces a useful concrete and tangible result is capable of patent protection in the US, but in Europe it is necessary to identify the technical contribution the invention makes. You will find some documents in Annex 3 on examples of "software concepts" that have been protected in Europe through patents.

However, it is beyond the scope of a short manual of this sort to deal with the differences in detail. A good patent agent will be able to advise you on this. ***Whatever your software invention is, it is useful to consider patent as a protection tool.***

Making the application

In making your patent application, it is essential to use the services of a patent agent. This is because both the drafting of the relevant documentation and the application process itself require a great deal of specialist skill. Applications can involve:

- filing for patent protection in all the major jurisdictions of the world because patent protection is territorial;
- translating the patent applications into local languages, and
- renewing your patents in the relevant countries at regular intervals.

All of these require specialist skill and can therefore involve considerable expense. In addition, if your patent is granted, you will have to pay renewal fees to the relevant patent offices. The frequency with which these need to be paid depends upon national law. Although they usually increase quite steeply on each renewal, this is unlikely to be a problem for most software inventions, because the life of the products tends to be short.

Things to watch out for: patents

Your first patent application filing only gives you priority rights to file a patent application in other member countries of the Paris Convention on Industrial Property for twelve months (most countries of the world). In other words, if you do not file for a similar patent in those other countries within a year of your first patent application, you may lose the possibility of protecting your invention by patent in those countries. Several regional and international patent systems exist to help you obtain international coverage by providing centralised patent procedures. The main systems used are the European Patent system administered by the European Patent Office and the Patent Co-operation Treaty system that is administered by the World Intellectual Property Organisation. You will find more information on these in Annex 3.

Formal protection: patents

If you fail to find venture capital or a partner within that time, and have not filed in the other jurisdictions you intend to target, you will be left with only one application, and that is in the jurisdiction in which you first filed. If someone from another country makes any infringement of your patent, you may run into legal complications and hence further expense.

If the patent route is appropriate for you, you must keep the software program information absolutely secret until you have filed the patent application. Employees working on software in a company should be bound to confidentiality before development commences. Outside consultants should also sign such agreements, and as with copyright, they should have agreed that the ownership of the rights in the software vests in you or your company. Remember too: a registered trade mark can be as valuable in protecting patented software as for protecting software that has not been patented.

Patents are relatively expensive to obtain. Although the initial filing in a single country is usually relatively cheap, the other costs associated with filing applications in other countries are substantial. They can also be expensive to defend legally.

High cost is often cited as the major barrier that prevents small firms from using patents. However, it may be possible to get the costs built into a venture capital agreement, and thereby spread the costs over a number of years.

Although this varies from jurisdiction to jurisdiction, patents are expensive to litigate, as provisional measures such as injunctions are rarely granted and the action usually is tried in full. Litigation insurance can be obtained, and some small firms find this helps, simply because the knowledge that you are backed by an insurance company frightens off would-be infringers.

a registered trade mark can be as valuable in protecting patented software as for protecting software that has not been patented

The cost of patents

high cost is often cited as the major barrier that prevents small firms from using patents

Annexes

Annex 1: registering a trade mark

In general, like patents, trade marks are registered by national authorities. Fortunately, international protection can be obtained through simplified international or regional procedures. The entire EU territory can be covered by a single "Community Trade mark" granted and administered by the Office for Harmonisation in the Internal Market in Alicante, Spain. Many important countries, including Japan (but not the US at the moment) belong to the Madrid System for the International Registration of Marks which allows a single application to be made for individual member states through the World Intellectual Property Organisation. Choosing between the different procedures will depend on the geographic coverage sought. International coverage is usually obtained by a combination of titles obtained through these various routes: national, European and international.

Obviously, the first thing you need to consider is the geographical protection you need. If the software is likely only to have a market in your own country because it is language or culture specific, it may be sufficient to have a purely national registration. But much software is intended to be marketed around the world, so you will need the widest possible protection. A trade mark attorney will advise you on this.

You can get a lot of useful information about this from the following websites –

The World Intellectual Property Organisation provides useful information about the protection of trade marks at the following address:

www.wipo.org/about-ip/en/trademarks.html

The Office for Harmonisation in the Internal Market administers the Community trade mark system. Information about this system can be obtained from:

www.oami.eu.int.

Annex 2: patent searching

This Annex gives details of places to find out about patent searching. The patenting of software has been allowed for some time, and there is quite a lot of 'prior art' about which you must try to discover. Your own expert knowledge of the industry, the products available and the scientific literature are the best guides to this.

Things to watch out for

There are several specific pitfalls to watch for in searching patent databases.

First, there is a delay between the filing date of a patent and its publication date. In the US, patents are at present only published on grant, although as explained in the main Patent section of this document, this is about to change. However, at the moment, there are rather a lot of 'unpublished' patents - those still being processed. In almost all other countries patent applications are published 18 months after their filing date.

A further problem is that the requirement for absolute global novelty when applying for a patent is not restricted to the information held in existing patents. For example, information that is published in books, discussed at conferences, and publicly available is equally capable of destroying novelty. This means that a patent search is not sufficient to detect all the relevant state of the art relating to a software invention. Patent searching requires specific skills and tools, and a free search on the Internet will only give you preliminary results.

www.european-patent-office.org/espacenet

This is a good place to start. Through the network which is accessible through esp@cenet, you will be able to search over 30 million patent documents world-wide.

www.ipr-helpdesk.org/espacenet

This site provides a comprehensive tutorial on patent searching.

www.european-patent-office.org/patlib

This site gives you access to the EPO network of patent libraries "PATLIB" that can offer you professional patent searching services .

If you would like to read more about patenting software, there is a lot of material to be found on the Internet, but the following websites are particularly recommendable.

www.european-patent-office.org

This site explains the European patent system

www.wipo.int/pct

This site explains the international Patent Co-operation Treaty system.

www.ipr-helpdesk.org/softpat

This site examines the trend of judicial reasoning in Europe and compares the situation in the US and Japan.

www.patent.gov.uk/snews/notices/practice/programs.html

This is a guide to the current position in Europe, published by the UK Patent Office.

www.softwarepatent.com.

This site contains some useful software patent resources.

Other useful sites include:

www.spi.org

www.european-patent-office.org/news/pressrel/2000_08_18_e.htm

www.european-patent-office.org/case_law/english/I_A_I-1.htm

www.ivanhoe.co.uk/books/content/ivpa/page234e.htm

The last two sites give you examples of software inventions that have been patented in Europe.

The following book is also recommended:

Intellectual Property Rights in Software: a Practical Guide.

This is published by the British Computer Society, ISBN 1902505-8-2.

Searching for patents

Annex 3: how to find out more about patenting

YOUR SOFTWARE

AND HOW TO PROTECT IT



This guide is based on research by experts on Intellectual Property Rights at the University of Sussex and the University of Sheffield.

The research was funded by Directorate General Enterprise of the European Commission.

