Do Microsoft Patents Apply?

A process for checking the patent safety of interoperable network protocols, including but not exclusively in the case of Open Source Software implementations.

Version 1.5
30th November 2013

Introduction: The Problem to Solve

Problem Specification: You have a network protocol you have implemented in code, and it is interoperable with some protocols that Microsoft implement. Now you would like to know if Microsoft is making any patent claims that read on your code, or conversely, if Microsoft expressly renounces any posssible claims they may have over the code.

Context: encumbering network protocols with patents has not historically been a winning commercial strategy. This document specifically covers the case of Microsoft protocols, where the issue is about Microsoft potentially using patents to protect its monopoly. There have been some cases that bear on Microsoft's behaviour with respect to patents on some specific network protocols, but Microsoft's behaviour is not constrained by any court in a general manner.

Scope: This paper only addresses implementation of protocols that Microsoft has documented. Most protocols of commercial significance or general interest that Microsoft implement are documented. The issue is no longer so much how these protocols work so much as what control Microsoft attempts to have over their commercial exploitation. The principle means of control Microsoft has is patents.

Table of Contents

Step 0: Get the Facts

Territories. What territories are you interested in? A patent can only be enforced in a territory where it has been granted, so if you don't care about a territory then it doesn't matter whether a patent has been granted or not. For practical reasons, any commercially relevant enquiry must assume that the EU and US are territories that matter, with others added as required.

Protocols. What is the list of protocols that you have implemented, using standard language as much as possible? Potential points of confusion and things to remember:

- Your opinon as to whether you have implemented a particular Microsoft patent-encumbered protocol may not match that of Microsoft. It is both the mechanisms in use when mediated by a network *and* the total effect of these mechanisms within the scope of the protocol which is covered.
- The mere fact of incompatibility with a Microsoft protocol is not proof of non-infringement. (nor vice-versa.) What does the patent say? How does the Microsoft implementation behave?
- It is usually possible to implement a protocol without infringing on a patent claim, by closely examining the text of the claim and considering what the workarounds are.
- Often in code it is technically convenient to implement two distinctly different but similar Microsoft protocols by sharing code or functions. This implementation convenience does not affect the material fact that more than two protocols have been implemented.
- Microsoft is careful to state that they reserve the right to assert patents over a non-Microsoft protocol implemented by Microsoft, such as a formally defined IETF Standards Track protocol. It may not be the most likely action, but Microsoft do not make any promise not to.

Step 1: Open Specification Promise and Open Exception

This step is about understanding Microsoft's statements about protocols, and seeing what applies to you. Be aware that Microsoft is using the word “Open” in a deliberately generic manner that has little bearing on how it is used in other contexts.

See http://www.microsoft.com/openspecifications/en/us/programs/osp/default.aspx for the Open Specification Promise (OSP), which includes a list of “Covered Specifications.”

All protocol specifications listed in the OSP are covered by a definitive non assertion of patent rights by Microsoft. You need to compare every protocol from your list from Step 0 and see if there is a match. You will often need to check the text of the protocol description to be sure. If all the protocols you implemented are covered by the OSP you can **stop now**.

Unfortunately, of the thousands of Microsoft protocols and formats in use on modern networks only a fraction are covered by OSP and chances are high that yours will not appear. So nearly everyone will need to go on to step 2.

The Open Exception comes from the Interoperability Principles, where Microsoft identify a list of Open Protocols they will only exert patents against if the patents appear on a particular list**.**

From the Interoperability Principles at http://www.microsoft.com/openspecifications/en/us/programs/interop/interoperability-principles/default.aspx :

> “Microsoft will make available a list of the specific Microsoft patents and patent applications that cover each protocol. We will make this list available once for each release of a high-volume product that includes Open Protocols. *Microsoft will not assert patents on any Open Protocol unless those patents appear on that list*.”

Emphasis added on the last sentence of this quote, because that is the Open Exception. We know from various places, including some cited below, that the definition of Open Protocol is the total of those protocols listed at http://msdn.microsoft.com/en-US/library/cc216514.aspx . The general comment Microsoft has about protocols is:

> Microsoft publishes technical specifications for protocols in the Windows client operating system (including the .NET Framework), Windows Server, Microsoft Office, SharePoint products and technologies, Microsoft Exchange Server, and Microsoft SQL Server that are used to communicate with other Microsoft software products. These specifications include Windows client operating system and Windows Server protocols covered by the MCPP and WSPP licensing programs created in accordance with the U.S. Consent Decree and the 2004 European Commission Decision.

This does not offer protection for all of these “Open Protocols”, it says that they “include” protocols covered by the WSPP program, but there are many others besides.

Also seemingly relevant is clause 1, “Open Protocols” of Interoperability Principle I:

> “Microsoft commits that all the protocols in its high-volume products that are used by any other Microsoft product will be made openly available to the developer community in a non-discriminatory fashion. These Open Protocols may include protocols that implement industry standards.”

This does not say that Microsoft will not assert its patent claims, this a version of the old RAND argument, which is known to be against interoperability and certainly against Open Source. So Principle I Clause 1 does not address the topic of this paper.

For Open Source Software there is clause 4 of Principle I, which states:

> Open Source Compatibility. Microsoft will covenant not to sue open source developers for development and non-commercial distribution of implementations of these Open Protocols.

While this appears to be a genuine covenant, it is very limited and does not address the topic of this paper. Microsoft is clearly targetting the common case where a group of open source developers create code which is then exploited by a company (which may employ some of those same developers.) Microsoft is merely promising not to sue individual developers for implementing code (such as network protocols) against which Microsoft asserts patent claims. However, it is true, individual open source developers should not expect to get individually sued for their work on patent grounds.

Step 2: Microsoft Patent Mapping Tool

See http://www.microsoft.com/openspecifications/en/us/programs/patent-map-tool/default.aspx .

This covers all of the protocols that Microsoft currently thinks may be relevant to patent discussions under particular Microsoft licensing programmes (including the Open Specification Promise covered in Step 0) although more protcols may be added at any time **and there may be errors**. Again, the total number of protocols here is only a fraction of all protocols in use, and does not cover all the protocols that Microsoft implements. Therefore it is very likely that the protocol you have implemented is not listed here.

Nevertheless it is worth a try!

Step 2 Background: How to Understand the Tool

Every way of using this tool gives results in the form of:

Patents	Patent Applications	Programs

The Patent gives a patent number, and especially a territory, such as “US”, or “Japan”. The territory is very important. The most common territory is the US.

Patent Applications disclosed potential patents, which may or may not be granted by the territory listed. This is useful to determine where Microsoft's patent claims may be going next on a particular topic.

The word “Programs” is confusing, and refers to a class of application rather than a licensing programme from Microsoft, although there is some overlap between the two. Examples of Programs are Exchange Protocols, PC Productivity Applications Protocols, and Windows Server Protocols.

The algorithm to use when interpreting territories is:

- No patent is listed (ie the tool says “None”). This means there will not be any patent in some other territory not covered by this tool. “None” is the best possible result.
- Patent listed in both the EU and the US: this is one that Microsoft really cares about, and probably has taken out in other territories. Keep looking, consult other sources than just Microsoft's tool or you may get a nasty surprise in a territory that matters (Japan is typically important for example, and Australia.)
- Patent is only listed in the US: that's relatively good news. There is a fair chance it will not exist elsewhere, with these two exceptions:
    - if there is an application listed in the US then watch carefully, because once it is granted, others may well follow in other territories, and
    - a very recent patent in the US – watch carefully, there is a process called IPC which gives a period in which Microsoft can apply in the EU.

Step 2a: Classes of Protocol Application

You may have implemented one or more protcols that fit within a particular class of application recognised by this tool, listed under the section called “Programs”. Click on this and you'll see classes of protocol grouped together according to the application type and/or the programme Microsoft wishes to offer them with.

For example, if you are a database developer, then you may choose “Interop – SQL Server Protocols”. That gives 49 results shown below, and you can see that it is relatively good news. Microsoft state that they have no patents that read on nearly all these protocols, a very small number have specific patents listed, and a larger but still fairly small number say “Available on Request”.

If you have identified a specific patent that Microsoft claims under a particular program that covers a protocol you have implemented you can **now go to step 3.**

Step 2b: Individual Patent Name

If you have not implemented a protocol from a class that Microsoft have listed, then you must scroll through the list, reading the title of the protocol. You may find a protocol you have implemented, because you recognise the name. More often, you must make a guess from the title and then you need to perform some research to see if it corresponds to a protocol you are interested in.

Be aware: there are likely to be cases where the title given by Microsoft in this tool is not something you recognise, and since Microsoft do not link to any descriptive text you are guessing. This is another reason why this tool is not definitive.

If you do your research and decide that, based on a review of protocol names and the protocol text that you believe corresponds to the protocol name, none of the listed protocols correspond to the work you have done, then you have exhausted what the Microsoft patent tool can do for you. **Now you can stop.**

Now you have a list of patents you can **go to step 3**.

Step 3: First Pass Examining the Patent Text

From one of the previous steps you have a list of patents that appear to be relevant to your work implementing protocols. Microsoft has patents which it claims covers protocols you think are the same or similar to protocols you have implement. Your list consists only of patents that Microsoft are claiming read on that protocol, and which they say they will enforce, noting the various exceptions given in steps 1 and 2. You have considered territorial implications, following the logic in Step 2 *Background: How to Understand the Tool*.

Congratulations, you know now exactly where Microsoft claims their patents apply to your work. From this point on you are in a standard patent evaluation scenario to see if Microsoft's claims are relevant, or if the patent is even valid, or perhaps to re-implement so that the question does not even arise.

Get a software patent lawyer!

A developer is not a software patent lawyer. A commercial lawyer is not a software patent lawyer, and neither is a patent lawyer.

Your task is to compare the technical descriptions in the patent with the technical description in your code, with the technical description in the Microsoft protocol specification. Your task is to compare three things which *might* express the same concept in very different ways, or may just seem to.

Good luck.