

FAQ: Licensing

라이선스

Mono로 상용 프로그램을 제작할 수 있습니까?

예. Mono를 이용해 상용 프로그램을 제작하려는 개발자들에게 라이선스 제도가 계획되어 있습니다.

Mono 프로젝트에서 사용된 라이선스는 무엇이 있습니까?

저희는 총 네 가지의 오픈 소스 라이선스를 사용합니다.

- C# 컴파일러는 MIT/X11 라이선스와 GNU GPL 모두가 적용되었습니다.
- 도구는 GNU GPL의 조항에 따라 릴리스 되었습니다.
- 런타임 라이브러리는 GNU 라이브러리 GPL 2.0 (LGPL 2.0)을 따릅니다.
- 클래스 라이브러리는 MIT X11 라이선스의 조항에 따라 릴리스 되었습니다.
- ASP.NET MVC, the Managed Extensibility Framework (MEF), the Dynamic Language Runtime (DLR), System.Data.Services.Client, parts of System.Numerics and ASP.NET AJAX client software는 오픈 소스 Microsoft Permissive License 아래 Microsoft가 릴리스 하였으며, 그 중 일부는 Apache2의 라이선스와 중복되는 것이 있습니다.

Mono는 LGPL, GPL, X11 조합이 당신에게 적합하지 않을 때 상용 라이선스를 받는 것이 가능합니다. Xamarin은 LGPL의 조항을 따르지 않는 Mono 응용 프로그램을 재배포하기 위한 상용 라이선스 옵션을 제공합니다. 당신 또는 당신의 조직이 Mono를 끼워 넣고 싶지만 GNU LGPL v2의 조항을 만족시키는 것이 불가능한 경우엔 Ultimate Edition이 적합할 것입니다. 다른 라이선스 옵션에 대한 정보를 받으려면 연락하십시오.

클래스 라이브러리가 MIT X11 라이선스를 따르는 이유는 무엇입니까?

원래 클래스 라이브러리는 LPGL의 조항에 따라 릴리스 되었습니다. GNU LGPL의 문제는 “연관된 일”에 대한 철 지난 용어였습니다. 라이브러리의 “연관된 일”은 라이브러리의 라이선스와 반드시 그 라이선스를 같이 해야 합니다. 이러한 정의는 객체 지향 프레임워크가 존재하기 전까지는 괜찮았지만, 객체 지향 프레임워크가 소개됨과 동시에, 객체 지향의 상속을 사용하는 몇몇 코드가 “연관된 일”의 한 종류인지에 대해 불만을 갖는 사람이 생겼습니다.

클래스 라이브러리는 Mono의 큰 구성 요소이며, 개개인의 다양한 그룹과 함께 발전해온 곳이기도 합니다.

이러한 모호함과, 모든 클래스 라이브러리 개발자들이 라이선스의 조항에 대해 서로 다르게 이해할 수 있는 가능성 때문에, 저희는 여전히 오픈 소스 및 프리 소프트웨어 라이선스를 따르면서도 이러한 모호함을 없앨 수 있는 라이선스를 채택하였습니다.

이러한 모호함은 코드 제작자가, 애플리케이션 개발자에게 기술에 의존한 소스 코드의 일부분을 릴리스 하도록 요구하게 만든다는 문제를 안고 있었고, 저희는 앞으로의 Mono가 이런 문제로 골치를 썩지 않았으면 했습니다.

Xamarin, Mono 런타임 응용 프로그램을 만들 때 라이선스는 언제 취득해야 합니까?

저희는 Mono와 임베디드 시스템에 기반한 Moonlight 또는 GNU LGPL의 의무를 충족할 수 없는 시스템을 사용할 때만 라이선스를 필요로 합니다.

예를 들면 당신이, 최종 사용자가 소스 코드로 Mono 가상 머신 또는 Moonlight 런타임의 업그레이드를 수행할 수 없는 장치를 제작한다면, Mono 및 Moonlight의 상용 라이선스가 필요합니다.

또는 당신이, Mono 런타임과 정적 연결을 필요로 하는 애플리케이션을 옮기면서 Mono에 다시 연결할 오브젝트 코드를 제공할 수 없는 상황이라면, 당신은 반드시 Xamarin의 상용 라이선스를 취득해야 합니다.

이에 대해 토론할 필요가 있다면 Contact form을 사용하십시오.

특정 라이선스 하에 Mono에 공헌을 하고 싶습니다. 어떤 라이선스를 인정하나요?

가장 먼저 호환성에 대한 라이선스를 평가할 것입니다만, 일반적인 규칙에 따라 “컨테이너” 모듈과 같은 조항을 따르는 코드를 받아들일 것입니다.

Why does Xamarin require a copyright assignment?

When a developer contributes code to the C# compiler or the Mono runtime engine, we require that the author grants Xamarin the right to relicense his/her contribution under other licensing terms.

This allows Xamarin to re-distribute the Mono source code to parties that might not want to use the GPL or LGPL versions of the code.

Particularly embedded system vendors obtain grants to the Mono runtime engine and modify it for their own purposes without having to release those changes back.

특히

Could patents be used to completely disable Mono?

No. For a more nuanced response, see the next question.

What kind of patent coverage exists for Mono?

The coverage is divided in various areas, that we explore below:

Code covered under the Microsoft Community Promise (ECMA/ISO standards)
Microsoft code licensed under the MS-PL or Apache2 Licenses, which come with patent grants.

Class libraries shipped with Mono

Code not in ECMA/ISO but opened with patent grants by the .NET Micro Framework.

Standards covered by the Microsoft Open Specification Promise

Code protected by the Moonlight 3/4 patent covenant

First some background information.

The .NET Framework is divided in two parts: the ECMA/ISO covered technologies and the other technologies developed on top of it like ADO.NET, ASP.NET and Windows.Forms.

Mono implements the ECMA/ISO covered parts, as well as being a project that aims to implement the higher level blocks like ASP.NET, ADO.NET and Windows.Forms.

The Mono project has gone beyond both of those components and has developed and integrated third party class libraries, the most important being: Debugging APIs, integration with the Gnome platform (Accessibility, Pango rendering, Gdk/Gtk, Glade, GnomeUI), Mozilla, OpenGL, extensive database support (Microsoft only supports a couple of providers out of the box, while Mono has support for 11 different providers), our POSIX integration libraries and finally the embedded API (used to add scripting to applications and host the CLI, or for example as an embedded runtime in Apache).

The core of the .NET Framework, and what has been patented by Microsoft falls under the ECMA/ISO submission.

Microsoft has announced that the ECMA standards for C# and the CLI are covered

under the Community Promise patent license.

Basically a grant is given to anyone who want to implement those components for free and for any purpose.

In addition to the grants made on the Community Promise, they have also released large swats of code under the Microsoft Public License and the Apache2 licenses, code that Mono bundles, for example the Managed Extensibility Framework, the Dynamic Language Runtime, ASP.NET MVC and even an embedded version of .NET and a complete CLI implementation with a complete class library called the .NET Micro Framework. All of the code in there comes with a patent grant.

The actual patent coverage includes the sum of both the ECMA-covered standards, their MS-PL licensed code and their Apache licensed code.

For people who need full compatibility with the Windows platform, Mono's strategy for dealing with any potential issues that might arise with ASP.NET, ADO.NET or Windows.Forms is: (1) work around the patent by using a different implementation technique that retains the API, but changes the mechanism; if that is not possible, we would (2) remove the pieces of code that were covered by those patents, and also (3) find prior art that would render the patent useless.

Not providing a patented capability would weaken the interoperability, but it would still provide the free software / open source software community with good development tools, which is the primary reason for developing Mono.

The patents do not apply in countries where software patents are not allowed.

For Linux server and desktop development, we only need the ECMA components, and things that we have developed (like Gtk#) or Apache integration.

Are there alternatives to ASP.NET?

Yes, there are a number of alternatives that people have been using for developing Web applications that do not use the ASP.NET model: Jitsu and Dream.

You can see Dream in action at Wik.is.

Are there alternatives to Windows.Forms?

Yes, there are plenty of alternatives, including Gtk# see our page on Gui Toolkits

for more information.

Is Mono only an implementation of the .NET Framework?

Mono implements both the .NET Framework, as well as plenty of class libraries that are either UNIX specific, Gnome specific, or that are not part of the .NET Framework but people find useful.

Mono can also be used as a Java Virtual Machine via IKVM.

Old Background

This contains some old information that we are keeping here for the sake of finding it for future reference. The information below is now superseded by Microsoft's announcement on using the Community Promise for the ECMA CLI and C#.

Jim Miller at Microsoft has made a statement on the patents covering ISO/ECMA, (he is one of the inventors listed in the patent): [here](#). Microsoft's letter to ECMA is [here](#).

Credits

The FAQ contains material contributed by Miguel de Icaza, Jaime Anguiano, Lluís Sánchez.