Programming in C# on Linux with Mono

1. Introduction

**What is Mono.NET ?**

Mono.NET is an open source implementation of Microsoft’s .NET framework. It was sponsored by Novell and now by Xamarin, and is based on the ECMA standards (links : <http://www.mono-project.com/docs/about-mono/languages/ecma/>) and the Common Language Runtime.

Basicly, it gives you the opportunity to create application in C# for multiplateform development (Android, IOS, Linux, Unix …). It is build to improve developper productivity.

Ther are four main componants that we can talk about Mono.NET :

* The C# compiler (wiki link : http://en.wikipedia.org/wiki/C\_Sharp\_%28programming\_language%29#Versions)
* Mono Runtime. The runtime provides a Just-in-time compiler an Ahead-of-Time compiler, a library loader, the garbage collector, a threading system and interoperability functionnality
* Mono Class Library. Obviously Mono is not just the Microsoft .NET library adapted for cross plateform applications. It has his own classes, for additionnal functionnality. For example : <examples>. And because it is open source, many users have implemented useful functions for Mono.NET.
* Base Class Library. The Mono platform has a solid foundation of classes compaptible with Microsoft .NET Framework classes to build applications on.

The big features of Mono.NET are :

* Multi-Language : you can developp with this platform in C#, Java, Python, VB 8, Ruby, F#, Eifel, Oxygene, and more ! Chose wisely !
* Multi-Plateform : the big improvement compare to Microsoft .NET : you can ru nit on Linux, Microsoft (of course), OS X, BSD and much more. You are free to code on any computer in any language ! (mostly)
* Open Source, Free Software : any one can contribute to it and improve it, help others and find solutions. The open source community is here to help !
* Binary Compatible : but only if built on an implementation of the ECMA’s Common Language Infrastructure (CLI) and C#. But it is still awsome.
* Comprehensive Technology Coverage : Mono.NET has taken the best and the most popular library and protocols to ad dit to its Framwork. So you won’t be lost.

**Why chosing Moono.NET ?**

It is popular : the Microsoft .NET is world wide famous, and Mono.NET hi sis prodigue son. It seduced millions of developpers, and millions of applications using Mono has been created, thousands of books, websites, exemple source code are available to help with any problem. And it is open source ! It is improving every day, every night.

The base class library is very helpful : if you need a protocol, object, ore anything (like a socket code or hashtable), you probably don’t have to write it yourself as it’s already built in !

It is a higher level progamming, you can concentrate on your application instead of writing system infrastructure code. Thanks to the runtime, automatic memory management, threading, reflection…   
And for those who tried to develop applications in other ways, they know it is not always pleasant.   
Mono.NET provides the opportunity to code in a high level language but also allow you to code woth very powerful development environements, like Visual Studio and Xamarin studio.

It is cross platform ! Like we said, it runs on Linux, Unix, Microsoft Windows, OS X, BSD, Sun Solaris, Sony Playstation 3, Nitendo Wii, Iphone, Android and many others ! You could run on nearly any computer in existance. Isn’t that beautiful ?

It uses the Common Language Runtime (CLR), that allows you to choose the language you are the most confortable with to code your application (in case you don’t like the C#). The list of language available : <http://www.mono-project.com/docs/about-mono/languages/>

Here we are going to develop in C#.

1. Installation

First think to say : installing a software on Linux is not easy.

Mono is not either, if you want to install it with the GTK# ressources. But binary installor are available for Linux and Windows !

Go to the Mono website and download it. Then, in the terminal, type :

$> tar xvfz mono-1.1.6.tar.gz

$> cd ./mono-1.1.6

$> ./configure

$> make

$> su

#> make install

#> exit

Then Mono is installed.  
To install GTK# :

$> tar xvfz gtk-sharp-1.9.2.tar.gz

$> cd gtk-sharp-1.9.2

$> ./configure

$> make

$> su

#> make install

#> exit

In case of GTK#’s configuration error, check the mono’s application’s configuration files (.pc) are installed in the right place.   
In our case, we had to copy them in /usr/lib/pkgconfig/ because the mono installation put them in /usr/local/lib/pkgconfig/

After the execution of all these commands, Mono is installed and configurated to use the managed WinForm and GTK#. SO, here we go !

1. An IDE for Mono : MonoDevelop

To develop C# applications with mono on Linux, you can use MonoDevelop, that is present in the binary installor of Mono for Linux.  
To install it, you have to get either the binary packets, the sources or the binary installor. Then follow the tutorial on MonoDevelop to compile and install it.

1. Consol programs with Mono

**Hello World**

That first program will show us the online compilation of the commandes and the Makefile that are in the program repository.

using System;

namespace Hello\_World //name of our project

{

public class Hello\_World //name of the class

{

public Hello\_World()//constructor

{

}

public static void Main(string[] args)// the main function

{

Console.WriteLine("Hello Linux World :-) ");

}

}

}

To compile this code with mcs, you have to type :

mcs -out:hello\_world.exe ./src/\*.cs

Explaination of this command :

|  |  |
| --- | --- |
| Command | Description |
| -reference  or –r | Referenced assembly in the program. The assembly has to be in the GAC |
| -lib or –l | Same but the assembly must not be in the GAC |
| -pkg | Bend the package with the program to compile (ex : GTK#) |
| -target or /t | Compilation’s target : exe, winexe, library, module |
| -out | Name of the output file |

This array is just an overview of the mcs compiler options. If you want to know them all, type :

mcs --help

or

man mcs

This is an exemple of the makefile we are using to compile with Linux/Mono :

all:

mcs -out:MonoMySqlIO.exe -r:./MySql.Data.dll -r:System.Data.dll ./MonoMySqlIO/\*.cs

echo "mono ./MonoMySqlIO.exe" > ./monoMySqlIO

chmod +x ./monoMySqlIO

The first line declare the name of the rule for the command.  
The second line do the compilation itself. Mcs is the C# Mono compiler.

-put :MonoSqlIO.exe says the name of the output file.  
-r :./MySql.Data.dll says we have to use the MySql.Data.dll assembly located in the current directory (./).  
./MonoMySqlIO/\*.cs says the location of the files for the compilation

The third line creates a file that allows to launch the program easily.  
echo « mono ./MonoMySqlIO.exe » allows to show the output on the standard output the text in « « .

1. Access to Mysql with Mono
2. IO with mono, upgrading the access to Mysql program
3. Conclusion