

# MONO

## A .NET Framework for Your Linux Systems

By Neale Ferguson



**T**he Holy Grail for developers, particularly Independent Software Vendors (ISVs), has been to write an application once and have it run anywhere. Such has been the promise of Java where, theoretically, an application can be created on one platform and run on another without modification. The developer also is provided with libraries of methods that take care of presentation, database communication, application serving, etc. Java environments are available for numerous platforms, ranging from z/OS to WinTel to embedded devices.

Microsoft also saw the advantages of a fully managed environment with a rich library of functions and came up with .NET. Naturally, .NET was restricted to Windows environments, which meant that ISVs developing for platforms such as Linux were unable to reuse work they may have undertaken for their Windows customers. The Mono project was started to remove that restriction.

### .NET and Mono: An Introduction

Before telling the story of Mono, it's appropriate to review the .NET technologies from Microsoft. First, .NET is

an umbrella term that applies to a collection of products, specifications, and technologies from Microsoft. The technology of relevance to Mono is the .NET Framework.

The Microsoft .NET Framework is a software component that's part of the Windows operating system that provides a library of classes designed to tackle common programming problems and a run-time environment known as the Common Language Runtime (CLR) under which these classes, and the programs written that use them, operate. Like Java, there's a virtual machine that manages the program and data. The CLR also provides other services such as security, memory management, and exception handling.

So where does Mono fit in? According to the Mono developers: "Mono provides the necessary software to develop and run .NET client and server applications on Linux, Solaris, Mac OS X, Windows, and Unix." More specifically, they describe Mono as being "... an open development initiative sponsored by Novell to develop an open source, Unix version of the Microsoft .NET development platform. Its objective is to enable Unix develop-

Supported Architectures	Operating System
S/390, S/390x	Linux
SPARC (32)	Solaris, Linux
PowerPC	Linux, Mac OS X
x86	Linux, FreeBSD, OpenBSD, NetBSD, Microsoft Windows, Solaris, OS X
x86-64: AMD64 and EM64T (64-bit)	Linux
IA64 Itanium2 (64-bit)	Linux
ARM: little and big endian	Linux (both the old and the new ABI)
Alpha	Linux
MIPS	Linux

Figure 1: Environments Supported

ers to build and deploy cross-platform .NET applications. The project implements various technologies developed by Microsoft that have now been submitted to the ECMA [European Computer Manufacturers Association] for standardization.”

Mono means “monkey” in Spanish and was developed by Ximian (another primate term) before they were acquired by Novell. Mono isn’t a product but a technology toolkit for developers with these features and characteristics:

**Multi-platform:** Unlike Microsoft .NET, Mono runs on many diverse platforms and architectures. Figure 1 shows the environments supported.

**Standards support:** The Common Language Infrastructure and the C# specifications are covered by ECMA specifications 334 and 335, which have been approved by the International Organization for Standardization (ISO).

**Multi-language:** Any compiler that produces “pure” Intermediate Language (IL) byte codes should work with Mono. The Mono project provides compilers for C#, Basic, and JScript. Supported languages include:

- **C#:** The main C# compiler of the Mono Project is mcs. There also is gmcs, which includes previews of several new C# features, such as generics and nullable types. In the future, gmcs will become the default compiler.
- **Java:** Mono can run Java code alongside .NET and lets Java and .NET objects interoperate. This occurs via KVM, which is the Java VM implementation that runs on top of .NET and Mono.
- **Boo:** This scripting language, similar to Python, is designed for the .NET framework.
- **Nemerle:** This is a new hybrid (functional, object-oriented, and imperative) programming language for the .NET platform.
- **Visual Basic.NET:** Full VB.NET support is available in Mono 1.2.3.
- **Python:** The first of two options is PythonNet. Brian Lloyd wrote a bridge to link the Python run-time with the .NET run-time. This uses the real Python engine and provides a bridge between the Python world and the .NET world to interoperate. The second option is IronPython. Jim Hugunin's compiler for Python is a complete implementation of Python from scratch that compiles Python code into native C Intermediate Language (CIL).

- **JavaScript:** Mono ships an implementation of the JavaScript language as part of its distribution; the main author behind this effort is César Natarén.
- **Oberon:** The Oberon compiler is written in Oberon.NET (another self-hosting compiler).
- **PHP:** Through the efforts of Raphael Romeikat, support from Thomas Uhl, and funding from Google’s Summer of Code 2005, a PHP Hypertext Preprocessor (PHP) compiler has been developed.
- **Object Pascal:** RemObjects ships an object Pascal compiler that’s supported in both .NET and Mono.

**Open source:** Parts of Mono are dual-licensed under GNU General Public License (GPL) or Lesser GPL (LGPL) forms, as well as under commercial terms from Novell for when the GPL and the LGPL aren’t suitable. To maximize adoption of Mono, there’s a tendency to use the MIT X11 or LGPL licenses.

**Commercially supported:** Novell offers commercial support for Mono.

**Comprehensive technology coverage:** The Mono suite consists of the components shown in the following section. Schematically, Mono can be depicted as shown in Figure 2.

### Mono Components

The following sections describe the status of the components of Mono.

### Mono

The base Mono component consists of the runtime, test suite, debugger, and documentation.

**Runtime:** The Mono runtime implements the ECMA Common Language Infrastructure (CLI). The Mono runtime implements this virtual machine

and provides a Just-in-Time (JIT) compiler, an Ahead-of-Time (AOT) compiler for selected platforms, a library loader, the garbage collector, a threading system, and interoperability functionality.

**Test Suite:** There are various kinds of tests in Mono, including Class Library Unit tests, Compiler tests, Runtime tests, and Web services tests.

**Debugger:** The debugger provides for: data inspection and manipulation, stack traces, control program execution, thread support, low-level state inspection, and process handling.

**Documentation:** The Mono documentation is divided into different sections available with the code or from the Mono Website—Class Library (API) Reference, Monkeyguide (The Mono Guide), resources to get started, articles, and man pages.

### Core Class Libraries

The core class library components provide the basic functions for a .NET application.

**Code Access Security (CAS):** There is preview support in 1.2 with full support in 2.0.

**Cryptography, Crimson:** Mono provides full .NET cryptography. Crimson offers a superset of the cryptographic features available in the .NET framework and is distributed separately.

**XML:** The System.XML.dll assembly is feature complete.

**Database Access:** Providers for open source databases include PostgreSQL, SQLite (Versions 2 and 3), Firebird Interbase, and MySQL. Providers for commercial databases include Mimer SQL, ODBC, Oracle (8i, 9i, and 10g), Microsoft SQL Server 7.0 (2000 and 2005), and Sybase ASE 12.0 and up. Commercially supported providers include VistaDB and OpenLink

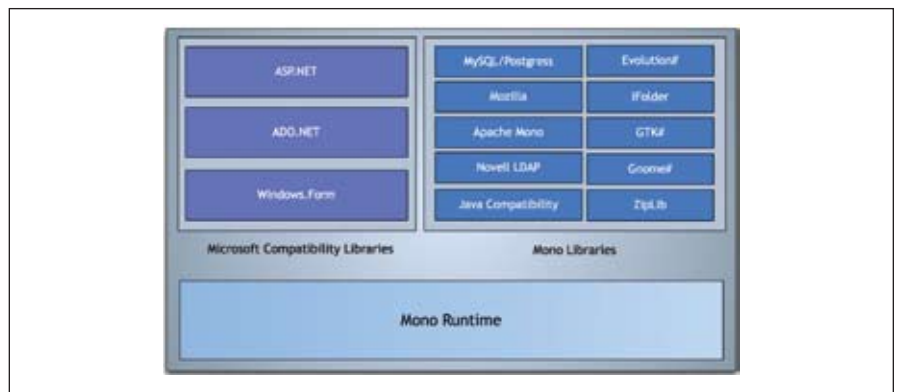


Figure 2: Mono Schematic

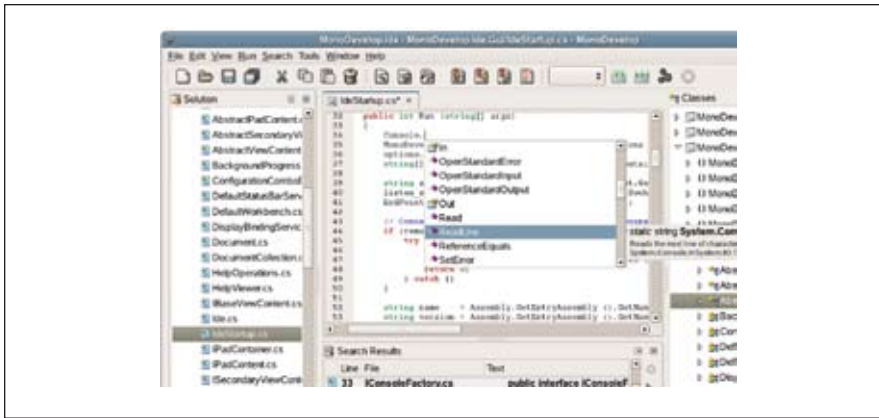


Figure 3: MonoDevelop Main Screen



Figure 4: Description of the Service Presented on the Browser for a User Request

```

///Web service that can make currency conversions
///</remarks>
[System.Web.Services.WebServiceBinding(Name="ConverterServiceSoap",Namespace="http://tempuri.org/"),
System.Diagnostics.DebuggerStepThroughAttribute(),
System.ComponentModel.DesignerCategoryAttribute("code")]
public class ConverterService : System.Web.Services.Protocols.SoapHttpClientProtocol {

    public UserInfo UserInfoValue;

    public ConverterService () {
        this.Url = "http://resc011/demo/ConverterService.asmx";
    }

    /// <remarks>
    ///Registers the user into the system
    ///</remarks>
[System.Web.Services.Protocols.SoapHeaderAttribute("UserInfoValue",Required=false,Direction=System.Web.Services.Protocols.
SoapHeaderDirection.Out),
System.Web.Services.Protocols.SoapDocumentMethodAttribute("http://tempuri.org/Login",RequestNamespace="http://tempuri.org/
",ResponseNamespace="http://tempuri.org/",ParameterStyle=System.Web.Services.Protocols.SoapParameterStyle.Wrapped,Use=System.Web.
Services.Description.SoapBindingUse.Literal)]
    public void Login(string a) {
        this.Invoke("Login", new object[] {
            a});
    }
}

```

Figure 5: Code Stub Generated by Web Service

Software. MonoQuery is an add-in to MonoDevelop, providing an SQL query and database browser tool. Other providers in Mono include OLE DB and IBM DB2 Universal Database.

**Mono Libraries**

The Mono libraries component provides extensions to the support provided with the core class libraries.

**Cecil:** A library used to generate, inspect, and modify other managed code.

**Security:** Secure Socket Layer, Transport Layer Security, Public Key Infrastructure, and Authenticode signing.

**Gtk#**

**Gtk# GUI Toolkit:** A GUI Toolkit is a set of .NET bindings for the gtk+ toolkit and assorted GNOME libraries. This library allows you to build fully native graphical Gnome applications using Mono.

**Stetic GUI Designer:** A new GUI designer for creating Gtk# applications.

**Gecko# Bindings (Mozilla):** A wrapper for the GtkEmbedMoz widget that provides the ability to embed the Mozilla rendering engine (Gecko) into your Gtk# application.

**OpenGL**

**Tao Framework:** A collection of

bindings and libraries to facilitate cross-platform, game-related development utilizing the Mono and .NET platforms.

**Gtk# and GL Integration:** GtkGLAreaSharp is an OpenGL development platform built on the Mono stack.

### Microsoft-Compatible Stack

**ADO.NET:** See the previous database access description.

**ASP.NET:** Mono's ASP.NET implementations support two kinds of applications: Web forms (Web Applications infrastructure) and Web services (the SOAP-based RPC system).

**ASP.NET Designer:** AspNetEdit is a visual design surface for ASP.NET, providing users with a RAD Web design tool on Linux. It allows users to manipulate ASP.NET Web Controls in an HTML designer.

**WinForms:** System.Windows.Forms (aka Managed.Windows.Forms) is one of the many GUI Toolkits for use with Mono, working toward compatibility with Microsoft's System.Windows.Forms.

**Enterprise Services / SystemMessaging:** Mono doesn't support either EnterpriseServices namespace or the SystemMessaging namespace.

**Transactions:** Volatile and durable resource managers are available and work with Single Phase Commit (SPC) and 2 Phase Commit (2PC).

**WSE:** Web services enhancements aren't supported.

**Olive - Beyond 2.0:** Olive is an unsupported set of add-on libraries and contains Workflow APIs, Communication Foundation APIs, System.Query, System.Xml.XLinq, and PresentationFramework and WindowsBase (only XAML support).

**Microsoft.Build:** Projects created in Visual Studio 2005 (except VC++2005) are saved in MSBuild format. XBuild is a Mono tool that facilitates the building of MSBuild files on Linux/Mono. This tool is still being enhanced.

**System.Query:** This library provides the necessary set of classes to obtain some features of C# 3.0 such as extension methods and query expressions.

### Developing Applications

There are at least two Integrated Development Environments (IDEs) for use with Mono:

**MonoDevelop:** This is an open source IDE primarily designed for C# and other .NET languages. Features include:

- **Code completion:** The IDE will attempt to complete type, method, and field names as they're typed. It also will automatically extract class information from your source code files and from the libraries referenced in the project.
- **Class management:** A class viewer that lets you list the classes in your project, their methods, and properties is included.
- **Add-ins:** MonoDevelop is extensible via the use of add-in modules.
- **Built-in help:** .NET and Gtk# documentation are built in.
- **Project support:** This tool comes with built-in projects to enable more rapid application development.

The latest version of MonoDevelop includes an option to directly import a Visual Studio .NET project. Figure 3 shows the main screen of MonoDevelop.

### Eclipse

There's a basic C# plug-in for Eclipse that implements auto indenting and syntax highlighting for this language. However, it lacks some of the more interesting features. Additionally, Eclipse can use the Gtk2 toolkit on GNU/Linux.

### Migration

While the goal of Mono is to provide the same level of support that .NET does in the Windows environment, the reality is Mono will always lag. This may create migration problems, depending on how leading (or bleeding) edge the existing .NET application is. A tool has been developed to assist in the migration process. Known as MoMA, this tool will analyze an application and flag potential problem areas. Detailed information on the areas flagged is provided so you can determine at a glance what the effort of any migration may be.

### Mono in Action

A multi-platform language and managed environment are a good start but are of no use unless people are actually developing and shipping applications. The following sections offer a sample of what's available and who is applying the specific tools.

### Commercial Applications

- Virtuoso Universal Server from OpenLink Software is a cross-platform

hosting environment for Mono/.NET/C#.

- Völcker used Mono .NET and C# to re-code its Active Entry software to run on Linux.
- Winfessor SoapBox Framework Mono Edition lets developers build .NET-based Instant Message (IM) solutions that run on any Mono-supported platform.
- Zamples lets software developers quickly learn new programming languages and Application Program Interfaces (APIs) such as C# and VB.NET by playing with live code examples using Mono.
- GMOVIL is a complete tool to manage common processes for Vodaphone sales points.
- Novell ZENworks Linux Management and iFolder 3.0 (on Linux and Mac OS X) are built using Mono.
- DocManager from NBFactory is a document management system.

### Open Source and Commercial Applications

- Banshee (formerly known as Sonance) is an application for music management and playback on Gnome.
- Beagle is a desktop search tool.
- F-spot is a photo management program from Novell.
- iFolder 3 is used for sharing files and folders by using a simple peer-to-peer fashion or Novell's groupware server products.
- Mainsoft bundles Mono libraries in its Visual Mainwin for Java 2 Enterprise Edition (J2EE) to deploy .NET Web applications on J2EE application servers.

### Open Source Applications

There are scores of open source applications written for Mono; some include:

- **MonoDevelop:** a port of Sharpdevelop to Mono and gtk#
- **MonoUML:** a Computer-Aided Software Engineering (CASE) tool
- **NAnt:** a free .NET development tool
- **OpenVPN-Admin:** a Graphical User Interface (GUI) client for OpenVPN written with Gtk#
- **SQL# for GTK#:** an SQL query editor and database browser for executing SQL statements and scripts; it runs on Mono and GTK# and works with MySQL, PostgreSQL, Firebird, Oracle, Sybase, Microsoft SQL Server, and Open Database Connectivity (ODBC)

- **Tomboy:** a desktop note-taking application that uses a wiki-like linking system
- **Lightweight Directory Access Protocol (LDAP) administration tool:** A GUI that will facilitate the maintenance of LDAP data sources.
- **Sussen:** a security scanner based on the Open Vulnerability Assessment Language.

Users of these tools include:

- **Novell:**
  - iFolder client and server on Linux and OSX
  - ZENworks Linux Management
  - Its Desktop search product Beagle
  - Photo management F-Spot
  - Groupware Server Hula
  - Media player software Banshee
  - Secret store manager (CASA)
- **Medsphere:** The Medsphere OpenVista software lets physicians access complete patient health information at the point of care; it runs on both Windows and Linux, thanks to Mono.
- **Völcker Informatik:** When customers began demanding Linux support for their ActiveEntry provisioning system, Völcker engineers turned to Mono to create a single cross-platform codebase.
- **Quantifi Solutions:** Specialists in complex financial instruments, Quantifi Solutions uses Mono to mix C# and C++ code in its finance modeling software. The company's finan-

cial models are constantly refined, so they need code that's easy to manage and maintain, while still running fast enough to keep up with the markets.

- **Otee:** Their Unity 3D game modeling tool uses Mono so customers can build cross-platform video games.
- **Versora:** They used Mono and C# to produce a cross-platform tool that helps companies move system and application settings and user data.
- **Fiducial:** Uses Mono to power its intranet.
- **Neoware:** Check the HTTP headers on the Website for thin-client vendor Neoware; those ASPX pages are served

up using mod\_mono, Apache, and Linux.

You also are using Mono if you're running SuSE Linux Enterprise Server (SLES) 10. The ZENworks software package manager is Mono-based.

### What's in It for System z?

Great looking and easy to develop GUI applications are well and good for the desktop and the laptop, but what roles can Mono play in the big end of town? Mono has function-rich, comprehensive libraries of XML and Web services methods that can be used to enable

the development of Service-Oriented Architecture (SOA) applications. In fact, companies such as Raris have SOA gateways that enable access to sources such as VSAM, DB2, and Adabas using programs written in C#.

Mono comes with a Web Services Description Language (WSDL) that will generate C# and VB.NET code that will enable applications to access Web services. Consider a simplistic Web service that provides a currency conversion service. Figure 4 shows a description of the service presented on a browser when the user requests information. Figure 5 shows how code may

**SOUND OFF !**  
 Comment on this article by visiting <http://community.zjournal.com/ApplicationDevelopment> and clicking on the thread titled "A .NET Framework for Your Linux Systems."

be generated for this service, which is the equivalent of issuing the command shown in Figure 6. This will generate the code shown in Figure 7, which then may be used by applications that wish to use the currency conversion Web service.

### Availability

Mono is now a standard part of SLES10 and SLED10. In fact, the ZENworks application used to manage packages on the system is written in Mono. If you're on another distribution of Linux, you can download the binaries from [www.mono-project.com](http://www.mono-project.com). **Z**

### References

- Migrating .NET Applications With Mono: [www.devx.com/racetolinux/Article/33945?trk=DXRSS\\_LATEST](http://www.devx.com/racetolinux/Article/33945?trk=DXRSS_LATEST)
- The Mono Project: [www.mono-project.com](http://www.mono-project.com)
- IKVM: [www.ikvm.net](http://www.ikvm.net)
- Nemerle: [www.nemerle.org](http://www.nemerle.org)
- PHP for Mono: <http://php4mono.sourceforge.net/>
- Object Pascal: [www.chromesville.com/](http://www.chromesville.com/)
- MoMa: [www.mono-project.com/MoMa](http://www.mono-project.com/MoMa)
- Raris: [www.raris.com/](http://www.raris.com/)

### About the Author

Neale Ferguson has more than 25 years of experience with IBM mainframe systems. He has been involved with Linux on the mainframe since the beginning. Originally from Sydney, Australia, he now works with Sine Nomine Associates in Northern Virginia.  
 Email: [neale@sinenomine.net](mailto:neale@sinenomine.net)  
 Website: [www.sinenomine.net](http://www.sinenomine.net)

```
> wsdl http://localhost/demo/ConverterService.asmx?wsdl
Mono Web Services Description Language Utility
Writing file 'ConverterService.cs'
```

Figure 6: Command to Generate Web Services Code Stub

Figure 7: Code Generated to Implement the Web Service