

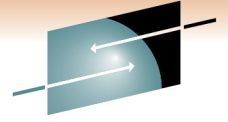
**SHARE**  
Technology • Connections • Results

# ASP.NET on zLinux: A New Workload

Mike Friesenegger  
Novell

Tuesday, March 1, 2011  
Session Number: 8473



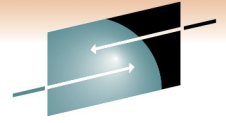


**SHARE**  
Technology • Connections • Results

# Agenda

- What is Mono?
- Suggested approach for a Mono proof-of-concept
- The Mono tools used for a POC and in production
- Wrap-up/Questions

# What is Mono®?



**SHARE**  
Technology • Connections • Results



Mono (mō•nō), *n*:

1. A complete and modern development platform for Linux
2. The necessary software to develop and run .NET client and server applications on Linux, Solaris, Mac OS X, Windows, and Unix
3. A thriving open source project with a growing community

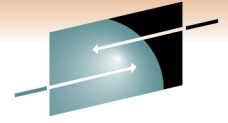
## What Can You Do with Mono?

Migrate Microsoft .NET desktop and server applications to Linux without significant investment in rewriting code

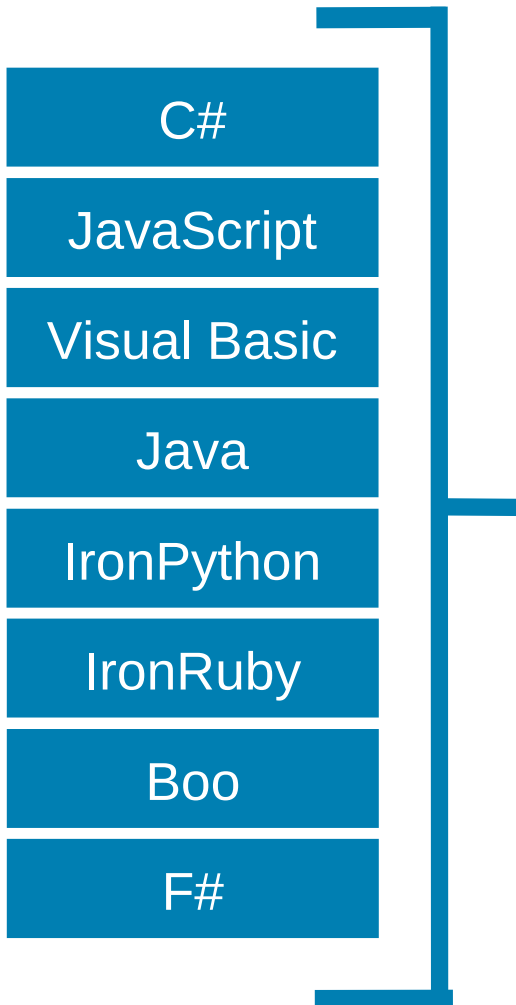
Target multiple platforms and increase addressable market

Leverage existing expertise in computer languages for more efficient development

# Multi-Language, Multi-Platform

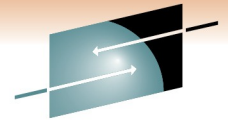


**SHARE**  
Technology • Connections • Results

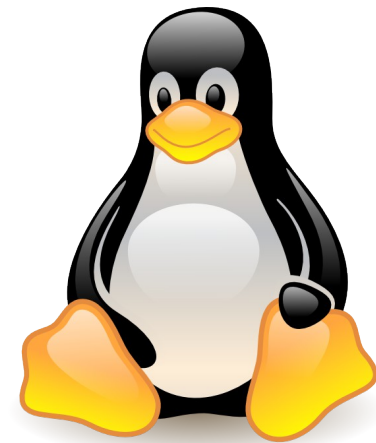


**SHARE**  
in Anaheim  
2011

# Binary Compatible

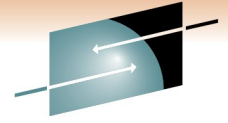


**SHARE**  
Technology • Connections • Results



**SHARE**  
in Anaheim  
2011

# Mono Today: Mono 2.10 in Feb. 2011



**S H A R E**  
Technology • Connections • Results

ASP.NET 1.1, 2.0 & 3.5, 4.0

Web Forms & Web Services

Supports ASP.NET AJAX and  
ASP.NET MVC3

ADO.NET 1.1 & 2.0

SQL Server, PostgreSQL, SQLite

Third parties supply providers for  
MySQL, Sybase, DB2, Oracle  
and more

LINQ including LINQ to SQL

Win Forms 1.1 & 2.0

C# 1.0, 2.0, 3.0, 4.0

New C# compiler backend

Upgraded VB compiler

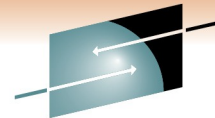
GTK#

Everything in .Net  
4.0, except WPF,  
EntityFramework,  
WF and limited WCF

Source: <http://mono-project.com/Compatibility>

Source: [http://mono-project.com/Release\\_Notes\\_Mono\\_2.10](http://mono-project.com/Release_Notes_Mono_2.10)

**SHARE**  
in Anaheim  
2011



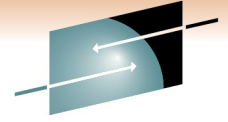
**S H A R E**

Technology • Connections • Results

# Suggested approach to a Mono Proof-of-Concept



# Novell Recommends a Phased Approach



**SHARE**  
Technology • Connections • Results

- Selling a proof-of-concept within your organization
- Select one or two ASP.NET applications to evaluate
- Prepare the Visual Studio developer environment
- Prepare the Mono server
- Functionality evaluation
- Performance evaluation



# Selling a Mono proof-of-concept within your organization

- Get executive level sponsorship
  - Development sponsor
  - Operations sponsor
- Suggest getting assistance from Novell or partner
- Outline the steps of the POC
- Define success criteria
  - Financial
  - Technical
- Set a proper expectation of POC duration
  - The POC can take three months or more depending on scope

# Select one or two ASP.NET applications for evaluation (1 of 3)

- Choosing the evaluation candidate
  - Poor
    - Windows.Forms desktop applications
      - *These depend heavily on Windows platform specific code which may be unavailable in Linux*
      - *May require additional porting effort*
    - Applications that rely heavily on the 3.0 "Foundation" Libraries - WCF, WPF, and WF
  - Good
    - ASP.NET applications (Web Forms and Web Services)
    - Other non-Windows desktop applications (console applications and other headless services)

# Select one or two ASP.NET applications for evaluation (2 of 3)



- Choosing the evaluation candidate
  - Best
    - Applications that meet the good criteria
    - A database which runs on Linux (especially zLinux)
      - *Mono has excellent support for MS SQL server*
      - *Far superior performance when database is on same machine as the application!*
- Run the Mono Migration Analyzer (MoMA) against the application
  - Submit the MoMA output to Novell for analysis

# Select one or two ASP.NET applications for evaluation (3 of 3)

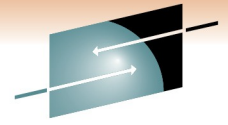


- **Optional:** Conference call with Novell or partner to review ASP.NET application selection
  - Review analysis of MoMA output
  - Discuss potential code changes that may be required for application to run on Mono
  - Discuss any third party components used by the application
  - Determine whether application is a good candidate for Mono POC

# Prepare the Mono developer environment\*\*

- On a Developer workstation
  - Install Mono Tools for Visual Studio
  - Install Mono for Windows
  - **Recommended:** Download and boot Linux target VM
    - Enable remote application debugging to Linux from within the Mono Tools in Visual Studio
  - Overview of Mono Tools functionality with developer
- Developer can begin testing ASP.NET application on local Mono instance

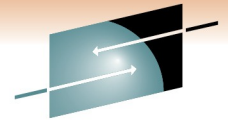
\*\* Contact Novell for the recommended version of Mono and Mono Tools to use during the POC



## Prepare the Mono server\*\*

- Install/configure z/VM (suggested for evaluation on System z)
- Install SUSE Linux Enterprise Server 11 using installation defaults
- Install the SLE Mono Extension
- Use BlogEngine.NET or some other free ASP.NET application to verify server is functioning
- **Optional:** Configure SAMBA share for developer(s) to publish application code

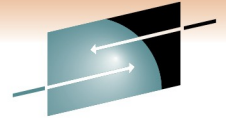
\*\* Contact Novell for the recommended version of the SUSE Linux Enterprise Mono Extension to use during the POC



# Functionality evaluation\*\*

- Make required code changes determined from analysis
- Deploy and test on Mono on Windows
- **Recommended:** Deploy and test on Linux target VM
- Deploy and test on Mono on SLES on System z
- Make any additional changes to resolve functionality issues

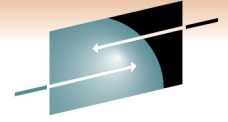
\*\* Evaluation should verify functionality, including database connections, user interfaces, integration with other applications, on each platform



# Performance evaluation

- Define acceptable performance metrics for application(s)
- Run application with load
  - Record application performance metrics
  - Collect Linux server performance metrics
- Work with Novell or partner for recommended changes to Linux server and Mono configuration
  - CPUs
  - Memory
  - Mono optimizations





**SHARE**  
Technology • Connections • Results

# The Mono tools used for a POC and in production



# The Mono Family

(Let's talk about the ones in **red**)



## Free

Mono Platform

MonoDevelop

Moonlight

MoMA: Mono Migration  
Analyzer

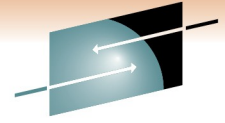
## Paid

SUSE Linux Enterprise  
Mono Extension

MonoTouch  
(Mono for iPhone)

Mono Tools for Visual  
Studio

# Mono Migration Analyzer (MoMA)



**SHARE**  
Technology • Connections • Results

- Inspects CIL byte code
- Looks for uses, compares with implementation
  - Missing methods
  - Unmanaged Calls
  - “Todo” methods (flagged with an attribute)

MoMA: Mono Migration Analyzer

mono migration analyzer

**Analysis Summary:**

- ✓ All methods called exist in Mono.
- ✓ No P/Invokes are called.
- ✓ No methods that throw NotImplementedException are called.
- ⚠ Methods called marked with [MonoTodo]: 91

MoMA Scan Results - Mozilla Firefox

File Edit View History Bookmarks Tools Help

file:///C:/Users/jhill/Downloads/MoMA\_24/Reports/output.html

**MoMA Scan Results**

Scan Date: 5/6/2009 7:55:16 PM  
MoMA Definitions: Mono 2.4 (Revised)

For descriptions of issues, see [MoMA Issue Descriptions](#).

Assembly	Version	Missing	Not Implemented	Todo	P/Invoke
✓ App_Code.dll	0.0.0.0	0	0	0	0
✓ App_global.asax.dll	0.0.0.0	0	0	0	0
✓ App_GlobalResources.dll	0.0.0.0	0	0	0	0
✓ App_Web_7cmx1ro6.dll	0.0.0.0	0	0	0	0
✓ App_Web_aemvubv5.dll	0.0.0.0	0	0	0	0
✓ App_Web_axk_w0es.dll	0.0.0.0	0	0	0	0
✓ App_Web_brglnpdg.dll	0.0.0.0	0	0	0	0
✓ App_Web_cwl-78eb.dll	0.0.0.0	0	0	0	0
✓ App_Web_dspzulkq.dll	0.0.0.0	0	0	0	0
✓ App_Web_ett-wzta.dll	0.0.0.0	0	0	0	0
⊟ App_Web_fcqaly2.dll	0.0.0.0	0	0	1	0

Version 2

Calling Method	Method with [MonoTodo]	Reason		
void rep_ItemDataBound (Object, RepeaterItemEventArgs)	string RegionInfo.get_DisplayName ()	DisplayName currently only returns the EnglishName		
✓ App_Web_gn0u-bgz.dll	0.0.0.0	0	0	0
✓ App_Web_huuwv8ub.dll	0.0.0.0	0	0	0

Done

Free download:

<http://www.mono-project.com/Moma>

**SHARE**  
in Anaheim  
2011

# SUSE Linux Enterprise Mono Extension

Based on Mono 2.6.7 release

Minimum requirements for installation & operation

- Same as SUSE Linux Enterprise Server

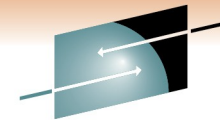
Supported processor platforms

- x86
- x86\_64 (AMD64 and Intel EM64T)
- IBM System z (64-bit)

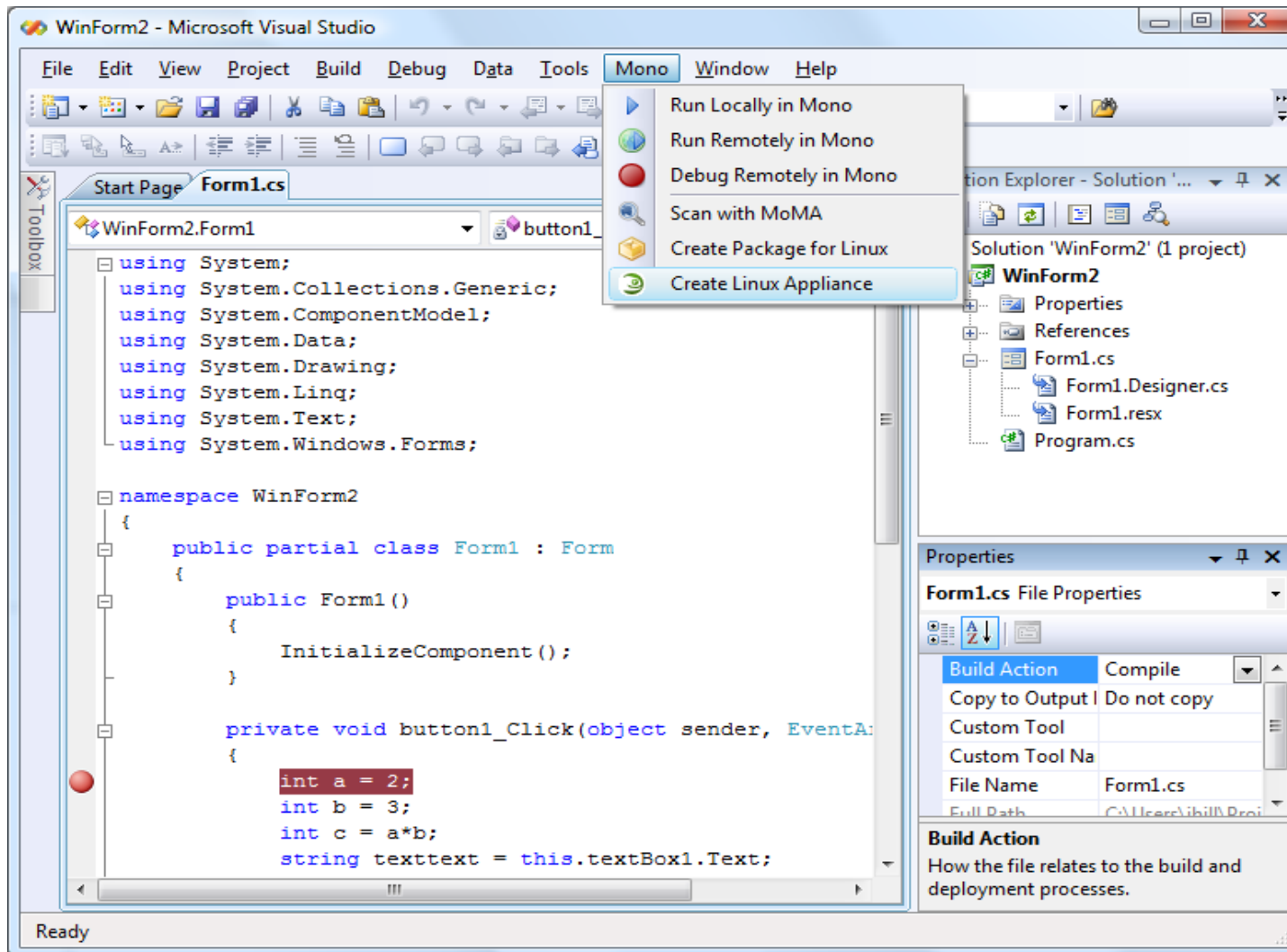


# Mono Tools for Visual Studio

## An Integrated Developer Experience



**SHARE**  
Technology • Connections • Results



The screenshot shows the Visual Studio IDE with the Mono menu open. The menu options are:

- Run Locally in Mono
- Run Remotely in Mono
- Debug Remotely in Mono
- Scan with MoMA
- Create Package for Linux
- Create Linux Appliance

The code editor shows the following C# code for Form1.cs:

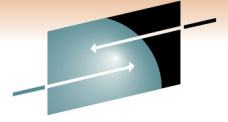
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WinForm2
{
    public partial class Form1 : Form
    {
        public Form1 ()
        {
            InitializeComponent ();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            int a = 2;
            int b = 3;
            int c = a*b;
            string texttext = this.textBox1.Text;
        }
    }
}
```

The Solution Explorer shows the project structure for WinForm2, including Form1.cs, Form1.Designer.cs, Form1.resx, and Program.cs. The Properties window shows the Build Action for Form1.cs set to Compile.

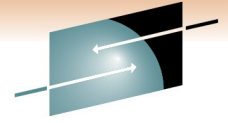
**SHARE**  
in Anaheim  
2011



**SHARE**  
Technology • Connections • Results

# For More Information

- Mono Project
  - <http://www.mono-project.com>
- SUSE Linux Enterprise Mono Extension
  - <http://www.novell.com/products/mono>
- Mono Migration Analyzer (MoMA)
  - <http://www.mono-project.com/MoMA>



**SHARE**  
Technology • Connections • Results

# THANK YOU FOR ATTENDING!!

