

Interoperability

between

MS-Windows, Other Operating Systems

And

Linux

Presentation Agenda

- **Background about Interoperability**
- **Problems with Interoperability**
- **Introduction and .NET framework**
- **Programming world – Basics**
- **Introduction to Mono and DotGNU**
- **Features of mono**
- **Installation of mono on Windows**
- **Installation of mono on Linux**
- **Mono supported Architectures**
- **Mono and DotGNU comparison**

Background

Need => Platform specific software such as;
Client Server / Desktop Applications
Web Applications
Standalone applications
Networking
Application services
Standards Need to work smoothly on multiple platforms

History of Interoperability => Wine
Samba
LDAP
POSIX
Cygwin- Linux like environment on windows
making it possible to port software running on POSIX such as Unix, BSD to windows
Virtualmachines/ coLinux

The Problem of Cross-Platform Portability

Ever since different processor architectures and different operating systems were developed, there have been problems with transporting software to different platforms due to:

- Differing processor instruction sets
- Differing system kernels and low-level libraries
- Differing programming languages, call sequences
- Differing binary and text formats for storing and transporting data and instructions
- Differing company and academic agendas
- The list goes on and on and on

Introduction

2002 Microsoft launched new system for platform independent development called “.Net (Dotnet) framework “

Microsoft has discontinued support for its previous favorite, the MFC library for Visual C++. All future Microsoft products will be based on the new technology.

Aim : To Simplify development of network and Internet applications
To support multiple languages with Unique Class libraries

Future: Despite the platform-independent approach of the .NET framework, Microsoft is highly unlikely to release a Linux implementation, and this lack of Linux support has prompted two projects to fill the gap

MONO and DotGNU

Introduction contd..

Both Mono and DotGNU are aiming to provide as complete an implementation of the .NET platform as possible.

The goal of both projects is to allow developers to exchange software across operating system boundaries.

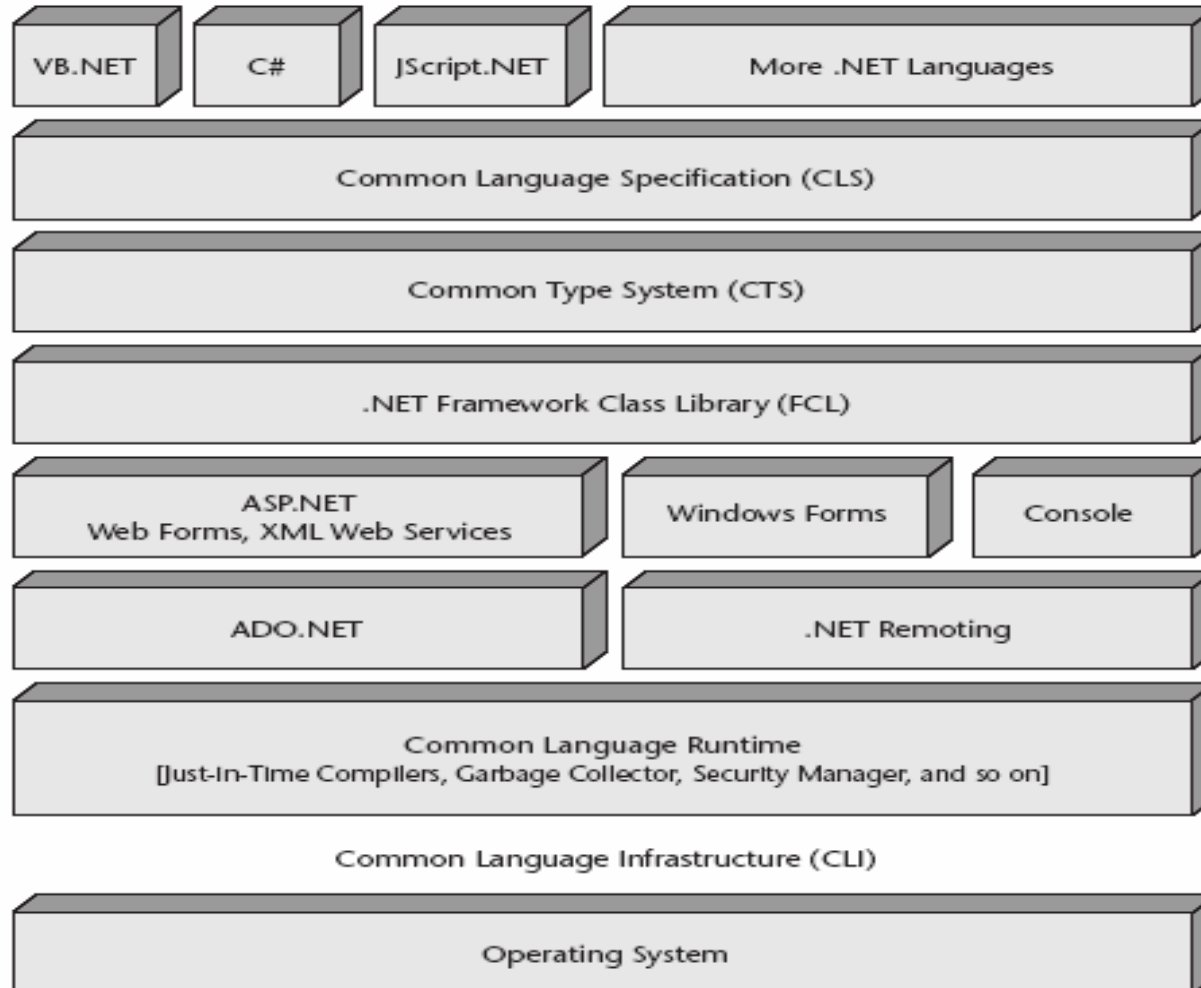
Besides basic functionality such as string manipulation, most projects need widgets for graphical user interfaces to help them cope with day-to-day tasks. Microsoft offers the `System.Windows.Forms` library for this purpose.

What is .NET?

- A Framework in which to run code
- A Common Language Runtime (CLR)
 - runs all programs
 - C# compiles to Microsoft Intermediate Language
 - MSIL runs on CLR
 - Virtual Machine like Java
 - code written in many languages compiles to MSIL
- A Common Language Specification (CLS)
- A Common Type System (CTS)

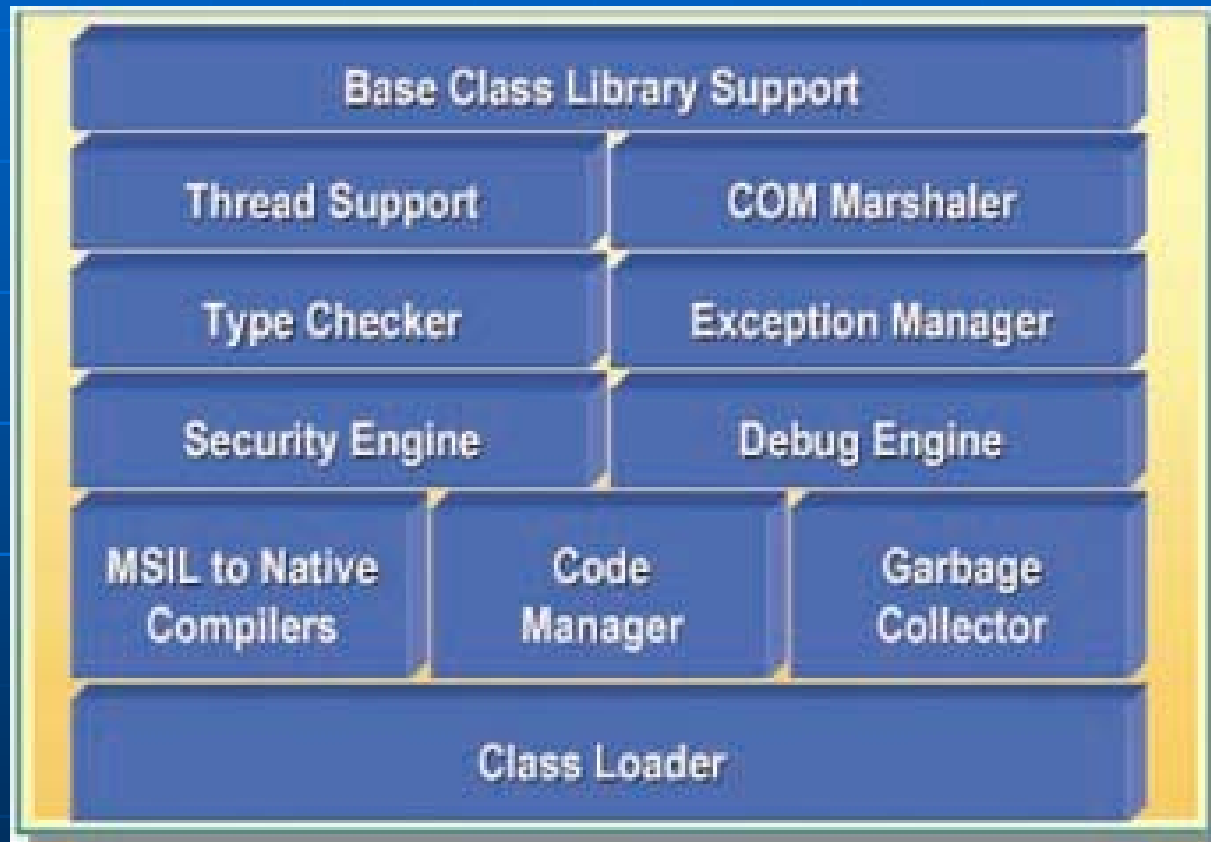
What is .NET?

Figure 1: An overview of the .NET architecture



Source: <http://www.windowsitlibrary.com>

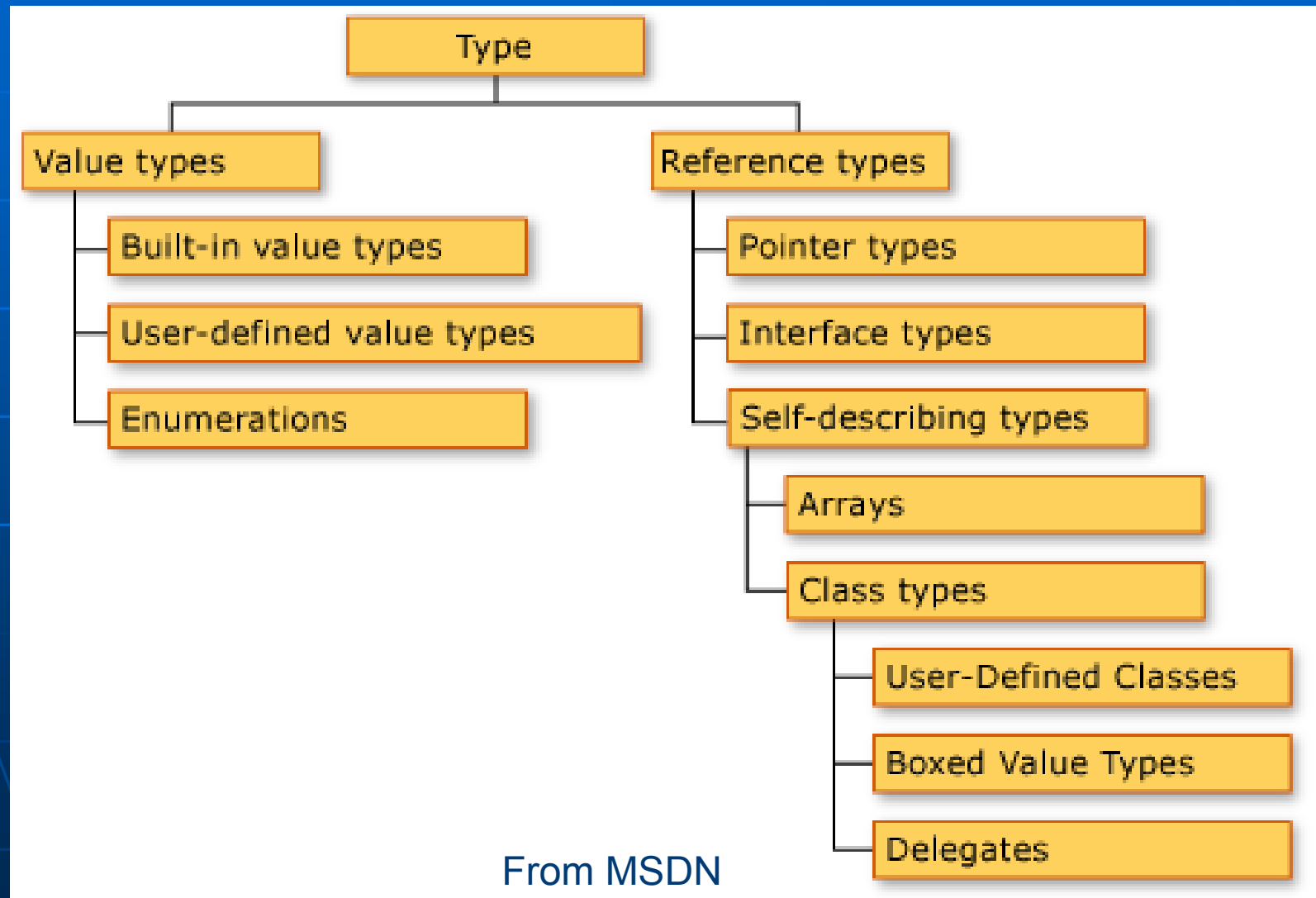
What is the CLR?



What is the CTS?

- A set of common types
 - any language that runs in CLR should implement
 - no syntax specified
 - Languages often define aliases
- For example
 - CTS defines `System.Int32` – 4 byte integer
 - C# defines `int` as an alias of `System.Int32`

What is the CTS?



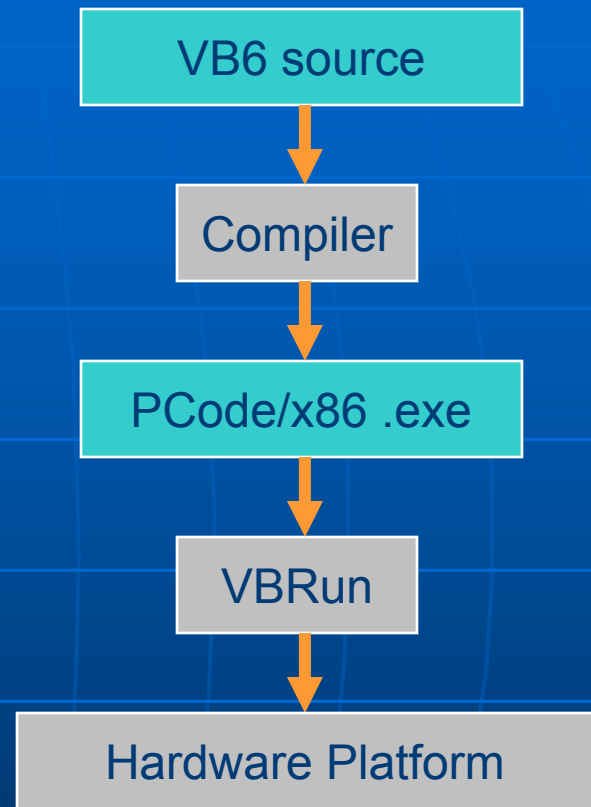
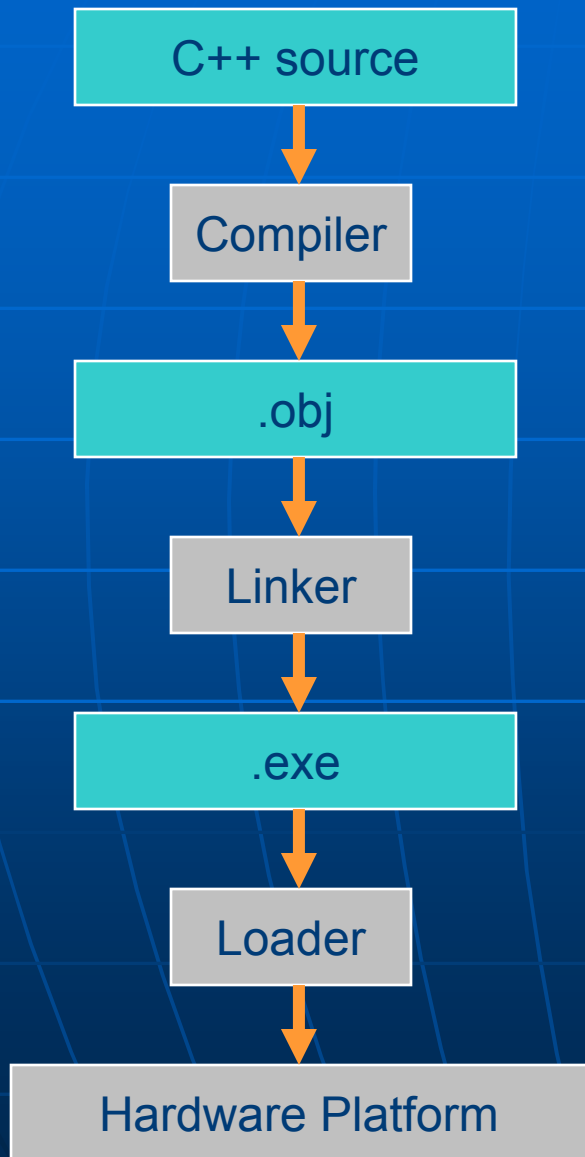
What is the CLS?

- A specification of language features
 - how methods may be called
 - when constructors are called
 - subset of the types in CTS are allowed
- For example
 - Code that takes UInt32 in a public method
 - UInt32 is not in the CLS
- Can mark classes as CLS-compliant
 - not marked is assumed to mean not compliant

The Class Libraries

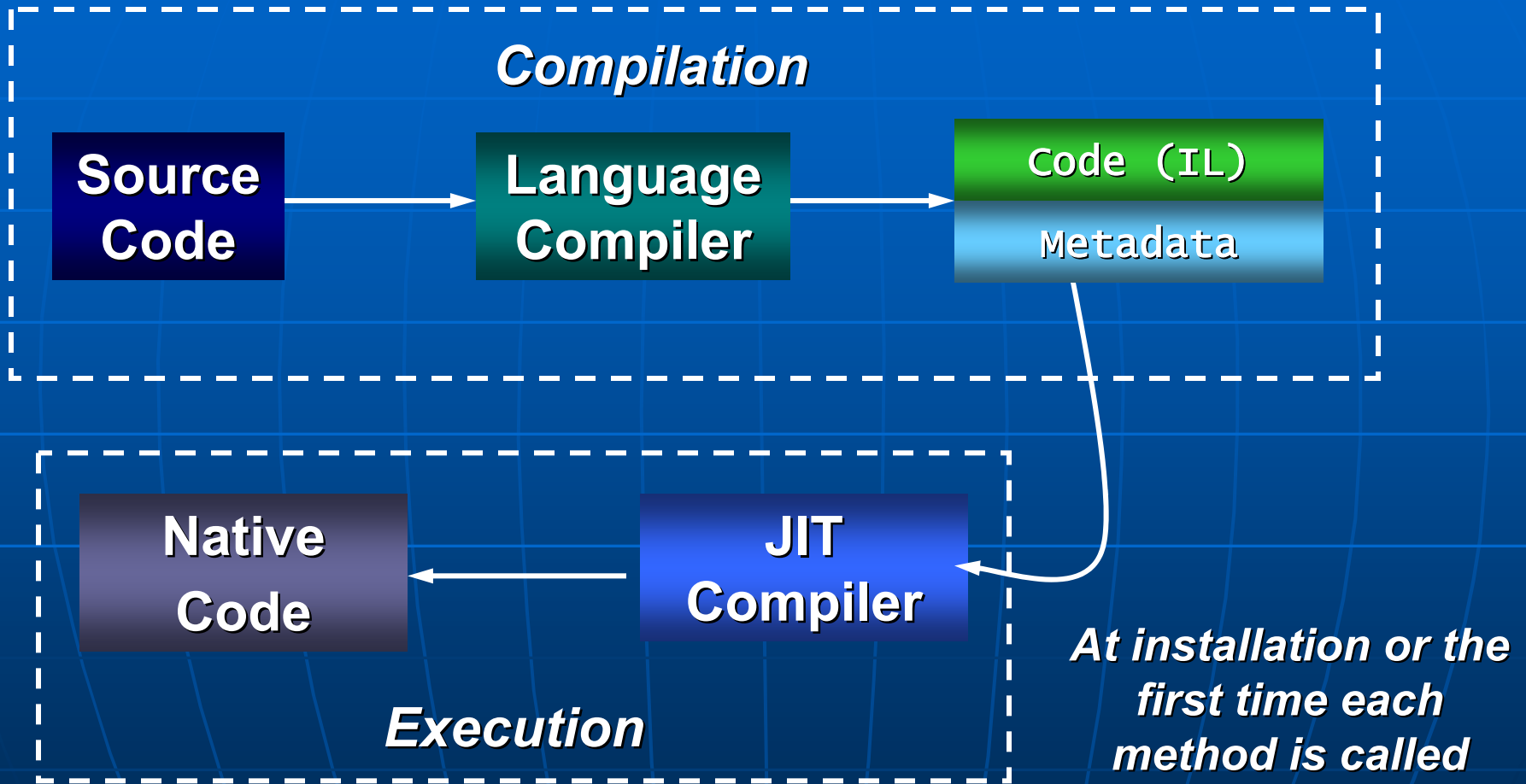
- The common classes used in many programs
 - like Java Class Library
 - eg.
 - `System.Console.WriteLine`
 - XML, Networking, Filesystem, Crypto, containers
 - Can inherit from many of these classes
- Many languages run on .NET framework
 - C#, C++, J#, Visual Basic
 - even have Python (see IronPython)

Basics: The Unmanaged World



Source: CLR 2005 Roadshow Slides

New Basics: The CLR



“Free Software” Definitions

“Free” software licenses grant:

1. the freedom to run the program for any purpose
2. the freedom to study and modify the program
3. the freedom to copy the program so you can help your neighbor
4. the freedom to improve the program, and release your improvements to the public, so that the whole community benefits

Freedoms 1 and 3 require source code access, because studying and modifying software without source code is extremely difficult and highly inefficient compared to modifying annotated source code.

- According to Richard Stallman of the Free Software Foundation

“Open Source” Definition On Line

An extensive definition of the related term “open source” is available at <http://www.opensource.org>

It speaks take in hand following points:

1. Free Distribution
2. Source Code
3. Derived Works
4. Integrity of the Author’s Source Code
5. No Discrimination Against Persons or Groups
6. No Discrimination Against Fields of Endeavor
7. Distribution of License
8. License Must Not Be Specific to a Product
9. License Must Not Restrict Other Software
10. License Must be Technology Neutral

Notable Examples of Free Software



Linux



FreeBSD



python



And many More!!!!

Introduction Mono

Implementation of the .NET CLI,CLR for GNU/Linux,



The Mono Project

- Began by the Ximian Company, founded by GNOME
- Novell Bought Ximian in August of 2004
- Mono is a platform for running and developing applications, based on the ECMA/ISO Standards (.Net and Java). Mono can run existing programs targeting the .NET or Java frameworks.”
- It is Free Software
- <http://www.go-mono.com> / <http://www.mono-project.com>
- Mono provides the necessary software to develop and run **.NET** client and server applications on **Linux, Solaris, Mac OS X, Windows, and Unix.**
- Uses XSP or Apache as its webserver.

Mono Features

- Multi-platform.
- Based on the ECMA/ISO standards.
- Runs ASP.NET and Winforms applications.
- Can run .NET, Java, Python and more.
- Open Source, Free Software.
- Commercially supported.
- Comprehensive technology coverage.

What .NET Technologies are Included in MONO?

Mono contains a number of components useful for building new software:

- A Common Language Infrastructure (CLI) virtual machine that contains:
 - a class loader
 - Just-in-time compiler,
 - and a garbage collecting runtime.
- A class library that can work with any language which works on the CLR.
- Both .NET compatible class libraries as well as Mono-provided class libraries are included.
- A compiler for the C# language.

Supported Architectures

Supported Architectures	Runtime	Operating system
<u>s390, s390x (32 and 64 bits)</u>	JIT	Linux
<u>SPARC (32)</u>	JIT	Solaris, Linux
<u>PowerPC</u>	JIT	Linux, Mac OSX
<u>x86</u>	JIT	Linux, FreeBSD, OpenBSD, NetBSD, Microsoft Windows, Solaris, OS X
<u>x86-64: AMD64 and EM64T</u> (64 bit)	JIT	Linux, Solaris
<u>IA64</u> Itanium2 (64 bit)	JIT	Linux
<u>ARM</u> : little and big endian	JIT	Linux (both the old and the new ABI)
<u>Alpha</u>	JIT	Linux
<u>MIPS</u>	JIT	Linux

Mono: Main Tools

- mono is the Mono runtime and Just In Time compiler (JIT)
- mcs, Mono's C# compiler
- gacutil is a tool used by developers to install versioned assemblies into the system Global Assembly Cache (GAC) to become part of the assemblies that are available for all applications at runtime.
- mint, the mono interpreter
- xsp, mono's stand alone ASP.NET web services and web application server
- mono-config - Mono runtime file format configuration

See: http://www.mono-project.com/Command-Line_Tools

What Else Does MONO Include?

- XSP: A Simple Web Server written in C# that can load ASP.NET 2.0 Apps
- Remoting.CORBA (<http://remoting-corba.sourceforge.net/>): A CORBA implementation for Mono.
- Ginzu: An implementation on top of Remoting for the ICE (<http://www.zeroc.com>) stack
- Gtk# (<http://gtk-sharp.sf.net>): Bindings for the popular Gtk+ GUI toolkit for UNIX and Windows systems. Other bindings are available: Diacanvas-Sharp and MrProject.
- #ZipLib (<http://www.icsharpcode.net/OpenSource/SharpZipLib/Default.aspx>): A library to manipulate various kinds of compressed files and archives (Zip and tar).
- GIGen (available from the Mono SVN): Bindings for OpenGL.

What Else Does MONO Include? Contd..

- Mono.LDAP: LDAP access for .NET apps.
- Mono.Data: We ship support for PostgreSQL, MySQL, Sybase, DB2, SqlLite, Tds (MS SQL server protocol) and Oracle databases.
- Mono.Cairo: Bindings for the Cairo (<http://www.cairographics.org>) rendering engine (Our System.Drawing is implemented on top of this).
- Mono.Posix: Bindings for building POSIX applications using C#.
- Mono.Http: Support for creating custom, embedded HTTP servers and common HTTP handlers for your applications.

Installation

- No prerequisites for installing Mono and XSP and Apache server.
- Downloads for Mono available for windows as an .exe file or for Linux as rpms.
- Download can be done from:
<https://project-mono.com/Downloads>

Installation

- Windows – Just run the .exe file
- Linux
 - Unzip the package downloaded
 - Change path to the unzipped folder
 - Install the individual rpms one by one by:
rpm -Uvh <filename>.rpm
 - If any dependencies arrive, try installing those rpms first on which other rpms depend

Compiling

- *For a C# code:*
 - *\$mcs <filename>*
- *For VB.Net code use:*
 - *\$vbnc <filename>*
- *For a Asp.Net file in C# use:*
 - *\$mcs /t:< desired target> /out:<output filename> - r:<reference assemblies list separated by comma/semicolon> <list of all .cs files in the project separated by spaces>*
- *For a Asp.Net file in VB use:*
 - *\$vbnc -l*

Execution

- For console or windows application:

- *\$mono <filename>*

Note that the file should be an executable file











- For web applications:

- *\$mono -r:<reference assemblies list separated by comma/semicolon> <start page name>*

Installation Steps

Windows

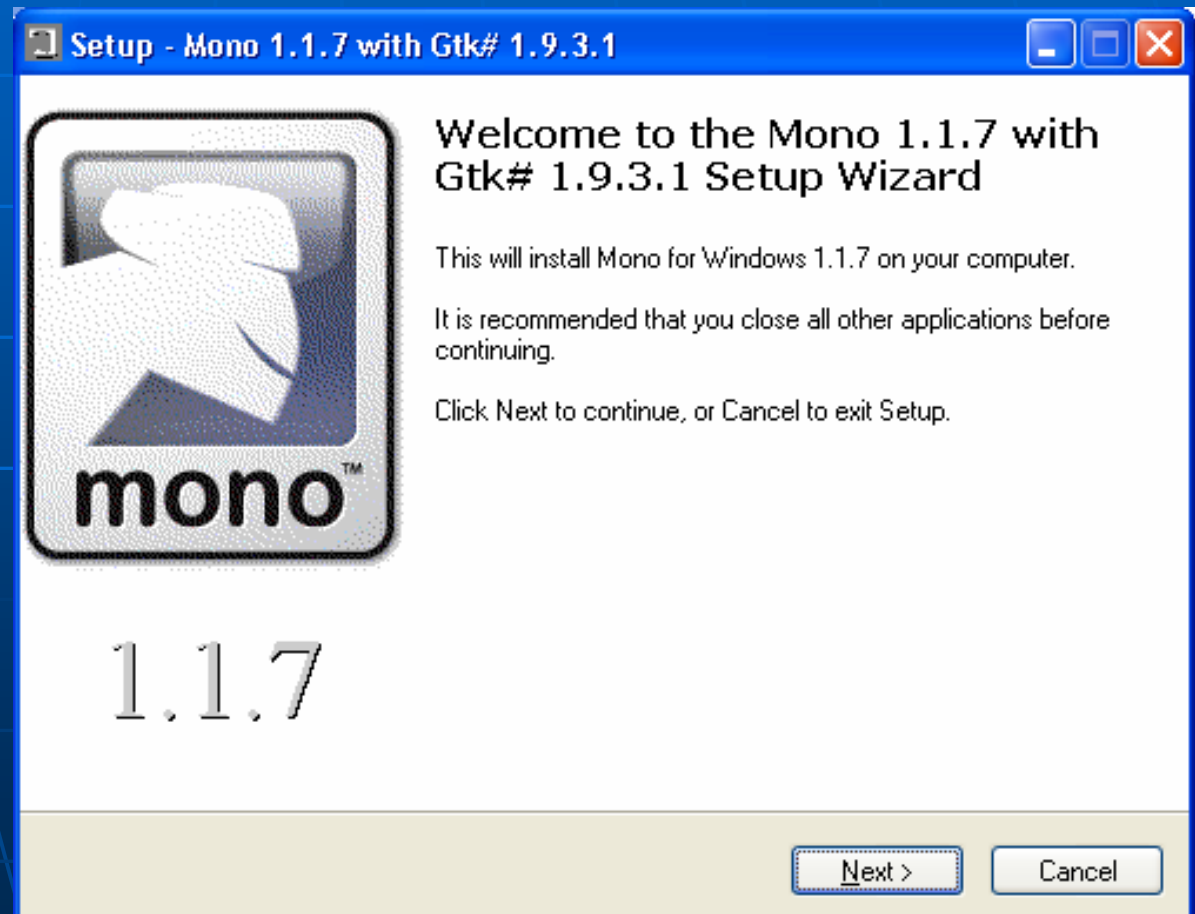
Step 1: Download

<> Source Code	Latest sources ↗
 Linux Installer for x86 (All distributions)	For help with the installation, see: Instructions to use the Installer. <ul style="list-style-type: none">■ Mono 1.1.7-1 Installer ↗
 Linux/x86	 SuSE Linux SUSE Linux Enterprise 9 x86 ↗ SUSE Linux Professional 9.2 x86 ↗ SUSE Linux Professional 9.3 x86 ↗ Novell Linux Desktop 9 x86 ↗  Red Hat Red Hat 9.0 x86 ↗ Fedora Core 3 x86 ↗ Red Hat Linux Enterprise 3.0 x86 ↗ Red Hat Linux Enterprise 4.0 x86 ↗
 Linux/x86_64	 SuSE Linux SUSE Linux Enterprise 9 x86_64 ↗ Novell Linux Desktop 9 x86_64 ↗
 Linux/S390	 SuSE Linux SuSE Linux Enterprise 9 for S390 (1.1.7) ↗
 Windows Installer Windows 2000 and above	Includes Mono, Gtk# and XSP: <ul style="list-style-type: none">■ Mono Setup (1.1.7) ↗
 Mac OS X	Includes Mono, Cocoa#, installs in /Library/Frameworks <ul style="list-style-type: none">■ MonoFramework-1.1.7.dmg ↗, for Mac OS X 10.3 (Panther) and Mac OS X 10.4 (Tiger)

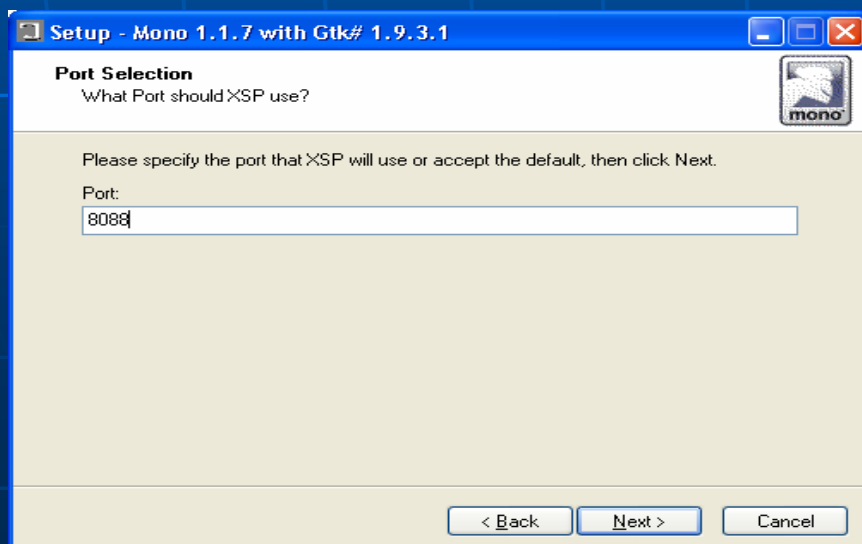
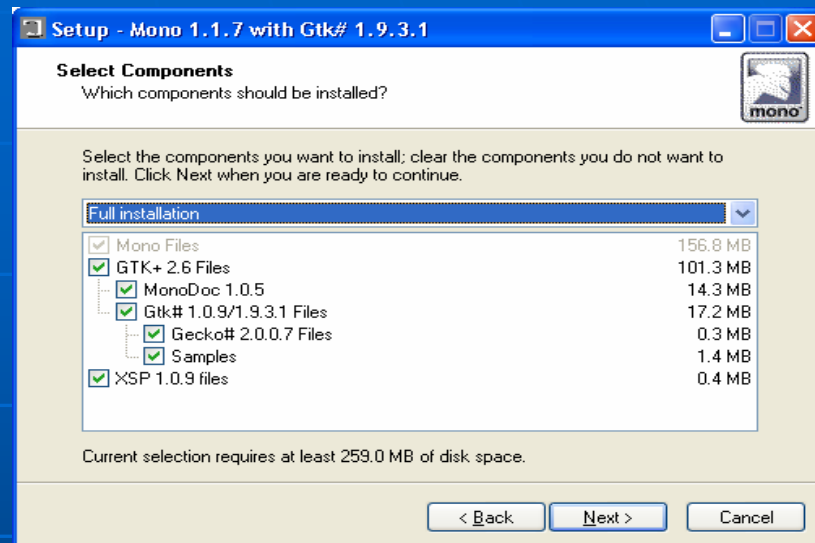
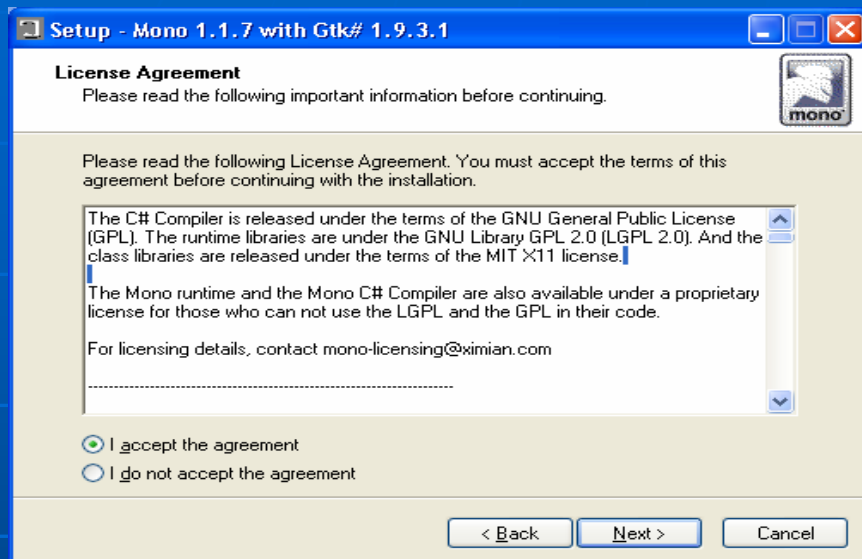
Installation Steps contd..

Windows

Step 2: Start Installation



Installation Steps contd..



Exploring MONO on Linux

To do this follow below steps;

- Make Linux Box ready (Redhat / SuSe)
- Select mono packages from the list available on the site
- Download either rpm or tar files
- You must know installation process using rpm or source files
- `man rpm` or follow installation procedure given along with source files.
- `Rpm -ivh filename`
- `Tar -xvzf filename -> ./configure -> make -> make install`

Some of the highlights of MONO Framework

- Core: mscorlib, System and System.XML assemblies. These support both the 1.x and 2.0 profiles. Work is underway to complete the 2.0 profile.
- ADO.NET: System.Data and various other database providers, they are 1.x complete, and most of 2.x is complete
- ASP.NET 1.x and 2.x: WebForms and Web Services are supported. Only WebParts are missing from our 2.x support.
- System.Security support 1.1 features and has partial support for 2.0 (like XML encryption) but the S.S.C.Pkcs namespace is still incomplete.

Some of the highlights of MONO Framework contd..

- DirectoryServices implemented on top of LDAP
- Windows.Forms 1.1 with partial 2.0 support.
- System.Drawing supports both 1.x and 2.0 profiles.
- Compilers: C# 1 and 2 as well as bits of 3, VB.NET 8 and various command line tools that are part of the SDK.
- Transaction support, partial support but currently no plans exist beyond the current implementation
- Open Source, Unix and Gnome specific libraries

Supported Architectures

Supported Architectures	Runtime	Operating system
<u>s390, s390x (32 and 64 bits)</u>	JIT	Linux
<u>SPARC (32)</u>	JIT	Solaris, Linux
<u>PowerPC</u>	JIT	Linux, Mac OSX
<u>x86</u>	JIT	Linux, FreeBSD, OpenBSD, NetBSD, Microsoft Windows, Solaris, OS X
<u>x86-64: AMD64 and EM64T</u> (64 bit)	JIT	Linux, Solaris
<u>IA64</u> Itanium2 (64 bit)	JIT	Linux
<u>ARM</u> : little and big endian	JIT	Linux (both the old and the new ABI)
<u>Alpha</u>	JIT	Linux
<u>MIPS</u>	JIT	Linux

Technologies Included in MONO

- A Common Language Infrastructure (CLI) virtual machine that contains a class loader, Just-in-time compiler, and a garbage collecting runtime.
- A class library that can work with any language which works on the CLR. Both .NET compatible class libraries as well as Mono-provided class libraries are included.
- A compiler for the C# language. In the future we might work on other compilers that target the Common Language Runtime.

What makes up Mono?

- A C# compiler.
- The Virtual Execution System: that has the Just-in-Time compiler, garbage collector, loader, threading engine.
- A byte code interpreter is provided for quickly porting Mono to new systems and debugging the JIT purposes
- An implementation of the .NET class libraries:
- Cross platform class libraries for data access:

What makes up Mono? Contd..

- UNIX class libraries: Mono.Posix
- Gnome-specific class libraries: The Gtk# family.
- Gtk# a toolkit to develop GNOME applications on UNIX and Windows.
- An implementation of the Remoting infrastructure in .NET

Companies Using Mono

- Novell (<http://www.novell.com>): Novell uses Mono for both client and server applications:
 - iFolder (<http://ifolder.com>) (client and server on Linux and OSX),
 - ZenWorks Linux Management (<http://www.novell.com/products/zenworks/linuxmanagement/>),
 - its Desktop search product (Beagle (<http://beaglewiki.org>)) etc.
 -
- MindTouch (<http://www.mindtouch.com/>): their Managed Office Server uses Mono under the hood as part of their Dream framework
- Fiducial (<http://www.fiducial.fr/>): This French business services company uses Mono to power their intranet.

Companies Using Mono contd ...

- Mainsoft (<http://www.mainsoft.com>) uses Mono for their Grasshopper product which allows ASP.NET applications to be deployed on J2EE servers
- Zing (<http://www.zing.net>): their portable MP3/Wifi player uses Mono to drive their application stack.
- Wikipedia (<http://wikipedia.org>): WikiPedia uses Mono for its search facilities. The indexing and the actual searching is done by Mono-based applications.
- GotMono.Net (<http://www.gotmono.net>)
- SplendidCRM (<http://www.splendidcrm.com>) Cross platform, CRM system.

Resources for Learning More

- Similar Projects

- DotGNU's Portable.NET: <http://www.dotgnu.org>

- Downloads

- <http://www.mono-project.com>
- <http://www.mono-live.com>
- <http://www.monoppix.com>
- <http://www.monodevelop.com>

Resources for Learning More

■ Web Sites and Books

- GtkSharp: <http://www.mono-project.com/GtkSharp>
- Novell: <http://www.novell.com/linux/>
- “Mono: A Developer’s Notebook”, by Niels Bornstein, Edd Dumbill: <http://www.oreilly.com/catalog/monoadn/>

■ IRC Chat Channels on irc.gimp.org

A lot of mono developers and core developers participate in these channels:

- #mono
- #monodevelop
- #mono-winforms
- #monodev

Conclusion

Mono is a useful and important project for a number of reasons:

- It provides a powerful platform for open source development across multiple platforms
- It provides developers the ability to leverage their skills across those different platforms
- It builds upon the same ISO and ECMA standards that the Microsoft .NET implementation builds upon and for that matter Java and J2EE.

Please visit the links referenced in the presentation, download, Install and code!

Thank you for your time!!