.NET Overview

David Smith

Microsoft Student Ambassador
CS Major
Michigan State University

Microsoft

Today's Topics

- Why I'm here.
- Exciting Demo – IssueVision
- What is .NET?
- Why learn .NET?
- Look into the Demo
- Old vs. New
- .NET Base Class Libraries
- ASP.NET vs. C#.NET
- Garbage Collection
- Visual Studio
- .NET User Group

Why am I here?

- For you
- It's a very exciting time to be a developer
- Very smart people here at MSU

What is .NET?

- A tool
- Microsoft® .NET is a set of software technologies for connecting information, people, systems, and devices
- Based on the ".NET Framework"

Microsoft's Vision for .NET

- OO Class-based Multi-language Programming Platform
- Targeting:
  - Multi-processor Servers to PDAs
  - Web-based Computing (XML)
  - Virtualization of Servers/TS

Microsoft's Vision for .NET

- Rapid Application Development (RAD)
- Common Platform for C#, VB, ASP, CE, XML, Visual C++, J#, Cobol/Pascal… from 1 Interface! (Visual Studio .NET)

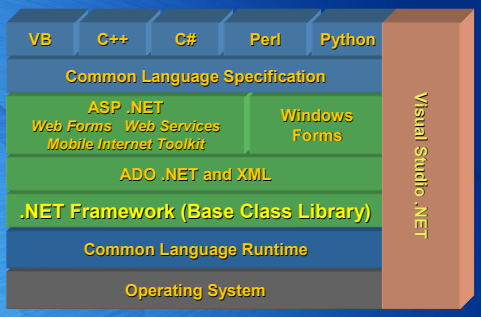
Why learn .NET?

- It makes your life easier
- Reduces development time
- More powerful applications
- The future of software evolution
- Service Oriented Architecture
- Valuable skill
- Huge base class framework to build off of

Demo Focus – A Look into IssueVision

- Smart Clients
- Versioning
- User Interfaces
- Web Services
- Authentication

.NET Framework



Current List of Language Compilers

- ADA
- APL
- Basic (Visual Basic)
- C#
- C
- C++
- Java Language
- COBOL
- Component Pascal (Queensland Univ of Tech)
- ECMAScript (JScript)
- Eiffel (Monash University)
- Haskell (Utrecht University)
- ICC (MS Research, Redmond)
- Mondrian (Utrecht)
- ML (MS Research Cambridge)
- Mercury (Melbourne U.)
- Oberon (Zurich University)
- Oz (Univ of Saarlandes)
- Perl
- Python
- Scheme (NW Univ.)
- SmallTalk

How is this possible?

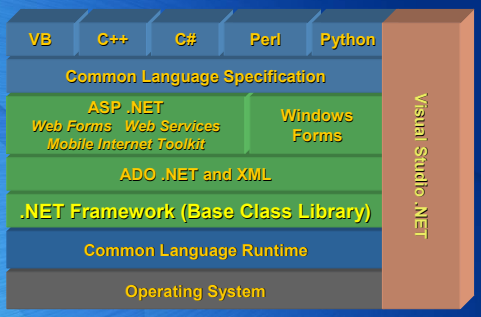
- .NET has 2 main components

1. Common Language Runtime (CLR) Microsoft Intermediate Language
2. .NET Framework class library

CLR- The Heart of .NET!

- Common Language Runtime (CLR)
  - Execution Engine for .NET Framework
  - Between the Binary .exe and the OS
  - Manages System Resources
    - Threads
    - Memory
    - Security

.NET Framework



Base classes and libraries

Very large base class framework

- System - Int32, String, Conversions, Basic Value Types
- System.IO - Files, Streams
- System.Collections - Arrays, Bit Arrays Stacks, Queues,
- System.Data - ADO.NET, SQL, XML, OLEDB
- System.EnterpriseServices - MSMQ
- System.Drawing - GDI+ (Graphic Device Interface)
- System.Reflection - Read object information at runtime
- System.Runtime - Remoting, Reflection

C#.NET vs ASP.NET Demo

C#.NET vs ASP.NET Demo

- Watch the similarities in code
- What makes this possible?
  - The .NET base classes

Web Services with XML

- Web Services allow extensible and flexible communication between services
- Promotes Service Oriented Architecture (SOA)
  - The next evolution of OOA

Old vs. New Demo

Old vs. New Demo

- Applying more power tools to class work and learning exercises.
- Teach students skills that will apply to real world disciplines
- More powerful tools allow developers to architect more realistic projects and libraries
- Analysis of large software design projects, less semantics of language

Garbage Collection

- No more pointers!
- Less semantics, more architecting
- Do what you love - build

.Visual Studio

- Most powerful IDE to enable developer productivity
- IntelliSense
- Made for developers
  - Word vs. Notepad
- Visual Studio 2005

Conclusions and Resources

- MSDN Academic Alliance
  - http://msdn.cse.msu.edu
- Microsoft Developers Network AA
  - Visual Studio .NET, XP, Server 2003, Visio, Project, SQL Server, MELL, Exchange…
- INETA – International .NET Association
  - http://www.ineta.org/

Programming Competitions

- The Imagine Cup
  - http://www.imaginecup.com
  - $25,000 in prizes
- http://cplan.cse.msu.edu/weblogs/
  - Full descriptions and documentations will be published

.NET User Group

- Goals:

1. Provide learning tools and resources for students interested in any MS software/technologies. (Not a class)
2. Compile a resource library of books on latest .NET technologies.
3. Get Guest speakers from Microsoft and INETA
4. Hold tech talks about latest software – a group learning environment.
5. Access to Microsoft's exclusive live web cast tutorial sessions on latest deployed technology.
6. Get an opportunity to interact with other groups and experts across the nation – People Networking.
7. Create, and apply the skills we've learned to new projects

- Questions for me?
  - Ask now!
  - Email: smith458@msu.edu

- If you're interested in the .NET User Group, or learning more about .NET
- Stay after, and join the conversation
- Change the world.
- Change the way people think.

ALP Session #2

David Smith

Microsoft Student Ambassador
Computer Science Major
Michigan State University

Microsoft

Today's Topics

- Classes
- Declarations
- Constructors
- Destructors and Disposal
- Methods
- Properties
- Polymorphism
- Inheritance
- Interfaces
- Abstract Classes

Classes: Why?

- Encapsulation
- Consolidation
- Reuse code
- Make changes once
- We should only have to solve a problem once

Classes: How?

- Keywords:
  - Public
    - Accessible to everyone
  - Private
    - Only accessible to the host object
  - Internal
    - Accessible to anyone in the assembly
  - Protected
    - Accessible to the host class and classes that inherit from the class

Constructors

- Demo Code

Destructors

- Demo Code
- Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

IDisposable

- IDisposable.Dispose()
  - Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

The GC is NOT predictable!!

- Finalizer: called by the Garbage Collector

```
~Class1() {
   // user code to handle destruction
   Dispose(false);
}
```

- Keep in mind:

  When Dispose(false) is called on an object, explicitly by the programmer, triggered by the Finalize()/~Destructor in a non-deterministic manner, the managed objects inside of class may have been destructed already, so we can not reference them

The GC is NOT predictable!!

```
Dispose ( bool disposing ) {
    if (!disposed) {

    // this is being called by the user
    // handle managed and unmanaged resources

     if ( disposing == true) {
              foreach ( IDisposable obj in this.coll)
                  obj.Dispose();
      }

     // Dispose was never called on this explicitly, only now by the GC
     // disposing == false; this is being called by the GC
     // We don't know if objects can be referenced anymore!
     // dispose of unmanaged objects ONLY

      else {
          this.unmanagedObject.Close();
      }

    disposed = true; // only enter this code loop once – ever.
    }
}
```

Classes: Adding Functionality

- Members
  - Public, private, protected, internal variables contained within a class
- Methods
  - Functions, actions
- Properties
  - Useful for containing controlled, and often more robust, explicit access to members

Members

- Demo code

Methods

- Console.WriteLine()
  "WriteLine" is the method
- Methods are verbs
- An action that a class can take
- A function that a programmer can invoke to process something

Methods

- Demo code

Properties

- Properties allow dynamic control over data access within a class

Properties

- Demo Code

Inheritance

- Code reuse
- Demo Code

Polymorphism

- Demo Code

Interfaces

- Public, published contract
- Often associated with policy
- No implementation code

Interfaces

- Demo Code

Abstract Classes

- Implementation + Inheritance
- Demo Code

Conclusions and Resources

- Students and faculty: MSDN Academic Alliance
  - http://msdn.cse.msu.edu
- http://google.com - Your best resource
- http://cplan.cse.msu.edu
  - The MSU Assoc. of Computing and Machinery Blog

- Questions for me?
  - Ask now!
  - Email: smith458@msu.edu

- If you're interested in the .NET User Group, or learning more about .NET
- Stay after, and join the conversation
- Change the world.
- Change the way people think.

ALP Session #3

David Smith

Microsoft Student Ambassador
Computer Science Major
Michigan State University

Microsoft

Today's Topics

- Networking / Remoting / Events Demo
- Delegates
- Events
- Asynchronous Callbacks
- Multithreading

Networking/Remoting/Events

- Musiciscrazy
  - Uses events to handle synchronization between clients: Publisher Subscriber model
  - Uses ASP.NET
  - Uses Remoting to communicate between clients and server

Networking/Remoting/Events

- Demo

Delegates

- A delegate is a managed pointer to a function
- A delegate matches a function signature
- // Declare delegate name and signature
  public delegate void StringLogging(string msg);
- // Declare a pointer to a function
  StringLogging stringHandler;

Delegates

- Demo 0

Problem solving with Events

- Scenario: Several objects would like to use some sort of notification to coordinate their actions.
- Problem: How can messaging be used to transmit events from one application to another?

Events

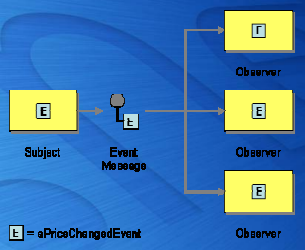
- Events are typically the defining concept of object oriented design
- The Object-Oriented Programming model both supports and relies upon events
- Events generally follow the publisher-subscriber pattern
- A publisher defines an event, and anybody that wants to know about that event can subscribe

Events

- Patterns that use Events
  - Event driven consumer

Events

- Patterns that use Events
  - Publisher-Subscriber



Events

- FakeEvents Demo
  - This is how it used to be

Events

- RealEvents Demo
  - This is how it is now

Events

- Demo of events with Interfaces
  - Establish public contracts that classes / webservices / programs use as a communication policy

Multithreading

- Run multiple threads at once to allow concurrent execution of code
- e.g. Run a GUI in the foreground and an indexing thread in the background

Multithreading

- System.Threading.Thread

Multithreading

- SimpleThreading Demo

Asynchronous Callbacks

- BeginInvoke()
- EndInvoke()

Conclusions and Resources

- Students and faculty: MSDN Academic Alliance
  - http://msdn.cse.msu.edu
- http://google.com - Your best resource
- http://cplan.cse.msu.edu
  - The MSU Assoc. of Computing and Machinery Blog

- Questions for me?
  - Ask now!
  - Email: smith458@msu.edu

- If you're interested in the .NET User Group, or learning more about .NET
- Stay after, and join the conversation
- Change the world.
- Change the way people think.

ALP Session #4

David Smith

Microsoft Student Ambassador
Computer Science Major
Michigan State University

Microsoft

Guidelines

- Do what you can, the best that you can, and wait for the results.
- Actions are all that matter

Today's Topics

- Generics & .NET 2.0
- WebServices
- Remoting
- Visual Studio 2005
- SQL Server 2005

Generics: Why?

- Speed
  - we save on not having to convert types at runtime
- Explicitness at design-time
  - We know that we'll be using a certain type – let's tell the code that
  - Runtime safety checks can be turned into design-time safety checks

Generics: How?

- class < type >
- Collection<string>

Generics

- Demo

Webservices: Why

- The next abstraction that is needed in order to make design possible
- Historically, this all makes sense
  - ASM -> C -> C++ > Java/C# -> Services -> ?
- Service based abstraction
- Service Oriented Architecture

Webservices: How

- The four tenets
  - Boundaries are explicit
  - Services are autonomous
  - Services share schema and contract, not class
  - Compatibility is based upon policy

WebServices: How?

- Musiciscrazy Demo

Remoting: Why?

- Remoting is a means of communication through instantiation of an object on a server

Remoting: How?

- System.Runtime.Remoting
- System.Runtime.Remoting.

Visual Studio 2005

- Design Time improvements
- Intellisense improvements

SQL Server 2005

- Fast
- Now Supports XML as a native type
- Integrates well with VS 2005

Conclusions and Resources

- Students and faculty: MSDN Academic Alliance
  - http://msdn.cse.msu.edu
- http://google.com - Your best resource
- http://cplan.cse.msu.edu
  - The MSU Assoc. of Computing and Machinery Blog

- Questions for me?
  - Ask now!
  - Email: smith458@msu.edu

- If you're interested in the .NET User Group, or learning more about .NET
- Stay after, and join the conversation
- Change the world.
- Change the way people think.