

Microsoft®
Visual Studio®
What's New For Coders

Daniel D'Agostino



MS Pre-Release Software

- Visual Studio 2010
- Windows Phone Developer Tools
 - XNA Game Studio 4
- Internet Explorer 9 Preview
- Office 2010



Agenda

- New IDE features
- .NET 4: C# meets VB
- Office Programming
- ASP .NET
- New Languages
- Windows Phone 7



About Daniel

- Web development since 2002
- BSc IT (Hons) graduate
- Software Developer at MITA
- Tech Spark member
 - Speaker Idol winner
- Game development enthusiast



New IDE Features



New IDE Features

- Multi monitor support
- Reference Highlighting
- Block Edits
- Navigate To
- Generate from Usage
- References Performance
- Intellisense Improvements
- Multi Targeting
- Starter Project Templates

Multi Monitor Support

The screenshot displays the Visual Studio IDE with multiple monitors. The main monitor shows the code for `Program.cs` in the `CSnew` namespace, featuring a `Program` class with a `Main` method. The secondary monitor shows the code for `Computer.cs` in the `CSnew` namespace, featuring a `Computer` class. The `Program` class in `Program.cs` includes the following code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace CSnew
{
    class Program
    {
        static void Main(string[] args)
        {
            // normal initialisation

            Computer laptop = new Computer("Laptop", "ASUS", "2500");

            // Optional parameters

            Computer junkserver = new Computer("JunkServer", "Gigabyte", "2500");
        }
    }
}
```

The `Computer` class in `Computer.cs` includes the following code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace CSnew
{
    class Computer
    {
        private String name = "";
        private String motherboard = "";
        private String harddisk = "";
        private String ram = "";
        private String cpu = "";

        public Computer(String name, String motherboard, String harddisk, String ram, String cpu)
        {
            this.name = name;
            this.motherboard = motherboard;
            this.harddisk = harddisk;
            this.ram = ram;
            this.cpu = cpu;
        }
    }
}
```

The `Program` class in `Program.cs` also includes the following code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace CSnew
{
    class Program
    {
        static void Main(string[] args)
        {
            // normal initialisation

            Computer laptop = new Computer("Laptop", "ASUS", "2500");

            // Optional parameters

            Computer junkserver = new Computer("JunkServer", "Gigabyte", "2500");
        }
    }
}
```

Reference Highlighting

```
Public Sub New(ByVal personName As String,  
               Optional ByVal personSurname As String = "")  
    Name = personName  
    Surname = personName As String  
  
    With hobbies  
        .Add(".NET")  
        .Add("Computers")  
    End With  
End Sub
```

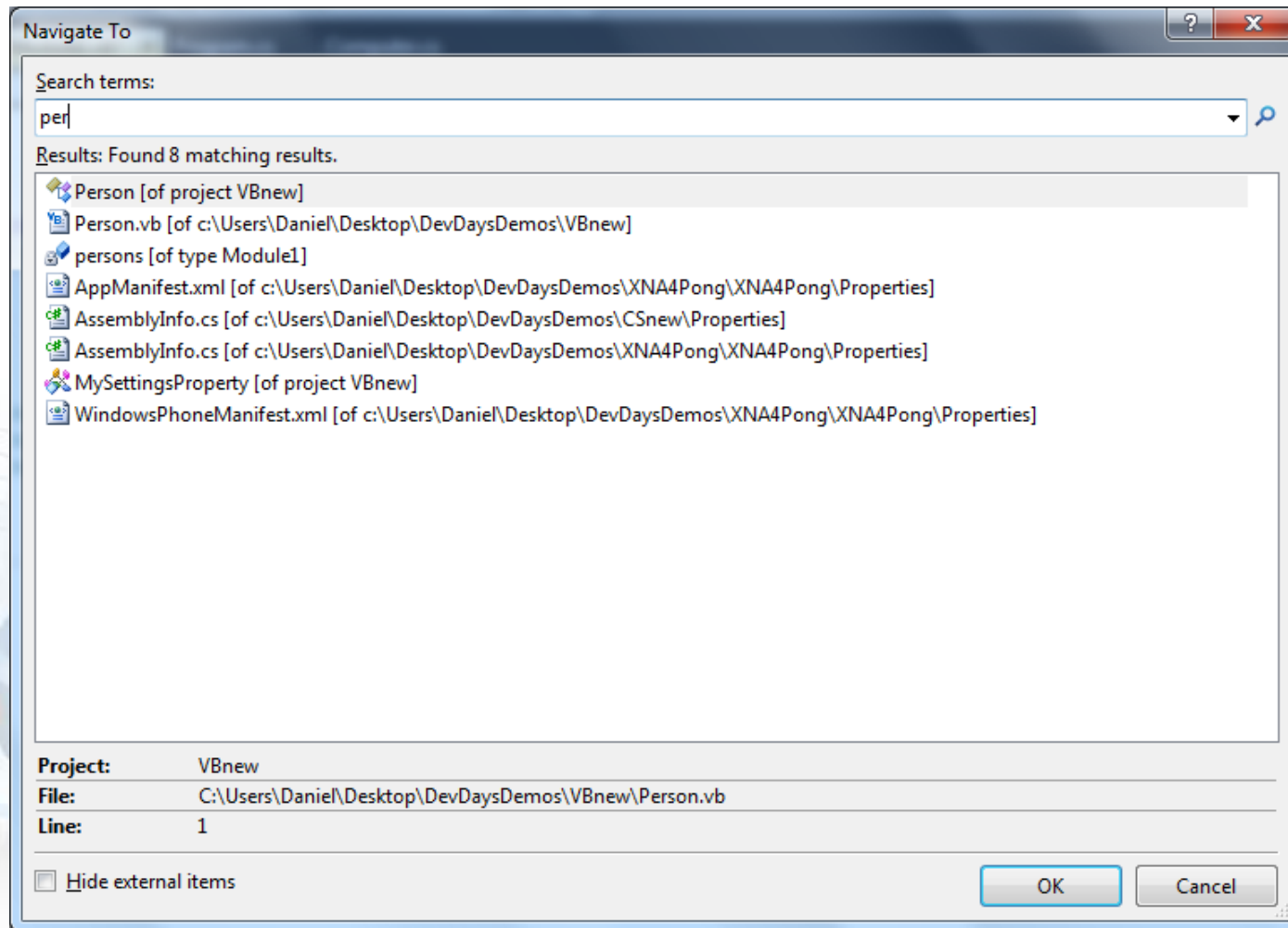


Block Edits

```
Dim john As Person = New Person("John")  
Dim mary As Person = New Person("Mary")  
Dim dave As Person = New Person("Dave")
```

```
Dim john As Person = New EmpPerson("John")  
Dim mary As Person = New EmpPerson("Mary")  
Dim dave As Person = New EmpPerson("Dave")
```

Navigate To



Generate from Usage

```
Dim dave As Person = New Person("Dave")  
  
dave.eat()  
End Sub  
  
End Module
```

! ▾

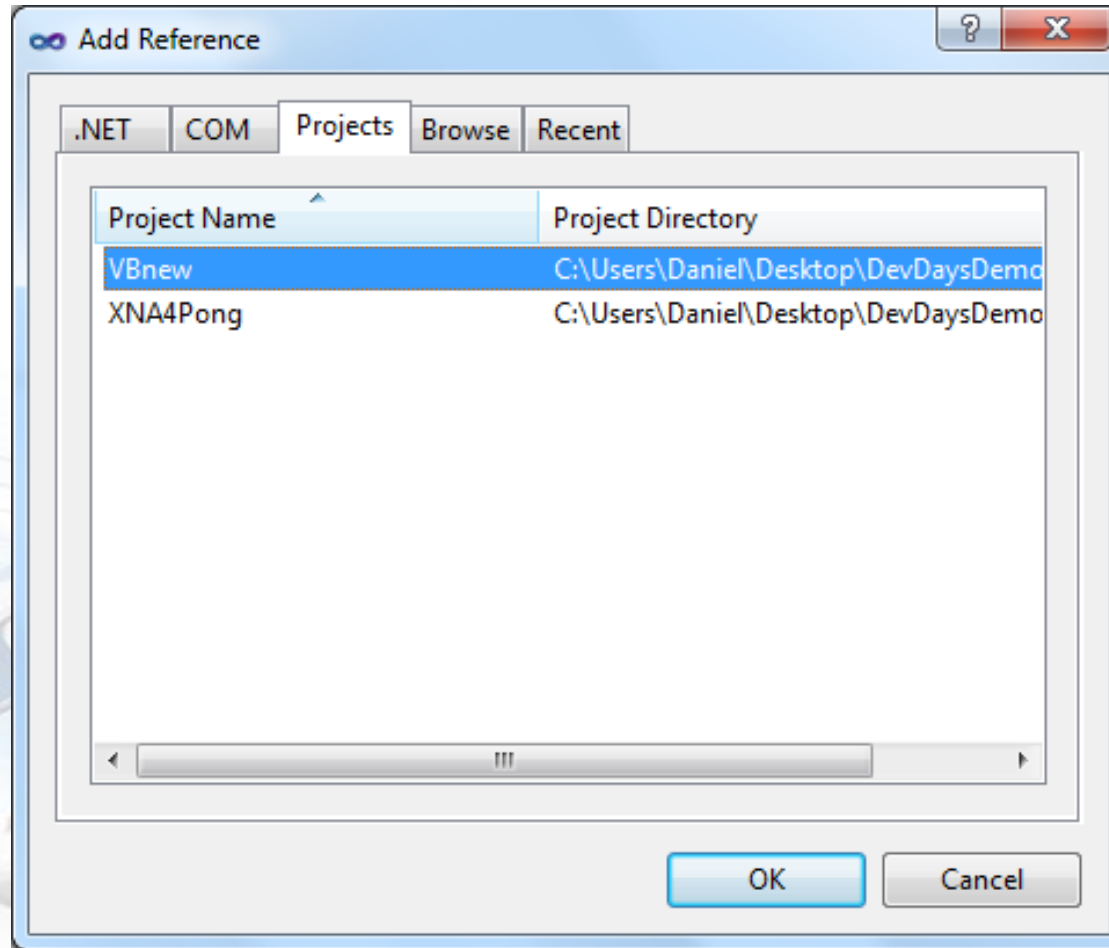
'eat' is not a member of 'VBnew.Person'.

Generate method stub for 'eat' in 'VBnew.Person'

Apply Fix

```
Sub eat()  
    Throw New NotImplementedException  
End Sub
```

References Performance



Intellisense Improvements

- Now searches for text in member names
 - As opposed to members beginning with text
- Pascal Case Intellisense

Person.e

End Sub

Module

Equals
ReferenceEquals

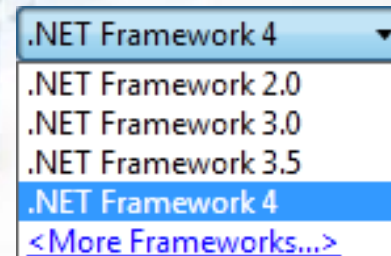
Person.re|()

End Sub

ReferenceEquals

Multi Targeting

- VS2008 supported targeting multiple versions of the .NET Framework
 - Intellisense in VS 2008 always shows methods for .NET 3.5
- VS2010 uses reference assemblies for each version of .NET
 - Contains only metadata for each assembly
 - Smaller assemblies
 - Accurate intellisense



Starter Project Templates

- Creating new projects
 - Can create empty project
 - Can also create project with basic functionality



.NET 4: C# meets VB



.NET 4: C# meets VB

- Auto-implemented properties (new to VB)
- Collection initialisers (new to VB)
- Lambda expressions (improved in VB)
- Multiline statements (new to VB, limited)
- Optional parameters (new to C#)
- Named parameters (new to C#)
- Covariance and Contravariance (new)

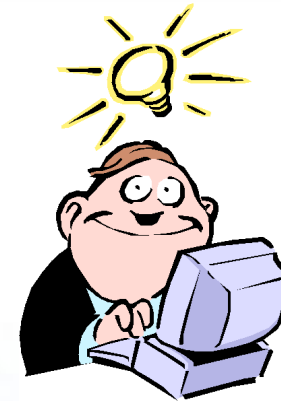
Auto-implemented Properties

' VS2008-style properties

```
Private _Name As String = ""
Property Name As String
    Get
        Return _Name
    End Get
    Set(ByVal value As String)
        _Name = Name
    End Set
End Property
```

' Auto-implemented properties

```
Property Surname As String = ""
```



Collection Initialisers (old way)

' Old way of initialising

```
Dim hobbies As New List(Of String)
```

' Constructor

```
Public Sub New(ByVal personName As String,  
               Optional ByVal personSurname As String = "")  
    Name = personName  
    Surname = personSurname  
  
    With hobbies  
        .Add(".NET")  
        .Add("Computers")  
    End With  
End Sub
```

Collection Initialisers (new way)

' Collection initialisers

```
Dim friends = New List(Of Person) From  
{  
    {New Person("Bill", "Gates")},  
    {New Person("Steve", "Jobs")},  
    {New Person("Tony", "Montana")}  
}
```



Lambda Expressions

- Concept: inline subroutines
 - E.g. $f := \lambda x . x + 1$

' Lambdas

```
Property gender As Char
Property title = Function(gender As Char)
    If gender = "M" Then
        Return "Mr."
    ElseIf gender = "F" Then
        Return "Ms."
    Else
        Return ""
    End If
End Function
```

Lambda Expressions

- VB Improvements in VS 2010
 - Multiline lambdas
 - Lambdas with no return value
 - Multiple statements



Multiline statements

```
Public Sub New(ByVal personName As String, _  
                Optional ByVal personSurname As String = "")
```

- Underscore no longer required for most statements
- Some exceptions

```
Dim friends = New List(Of Person) From  
{  
    {New Person("Bill", "Gates")},  
    {New Person("Steve", "Jobs")},  
    {New Person("Tony", "Montana")},  
}
```

Expression expected.



Optional Parameters: Why?

HFONT CreateFont(

```
    int nHeight,           // logical height of font
    int nWidth,            // logical average character width
    int nEscapement,       // angle of escapement
    int nOrientation,     // base-line orientation angle
    int fnWeight,         // font weight
    DWORD fdwItalic,      // italic attribute flag
    DWORD fdwUnderline,   // underline attribute flag
    DWORD fdwStrikeOut,   // strikeout attribute flag
    DWORD fdwCharSet,     // character set identifier
    DWORD fdwOutputPrecision, // output precision
    DWORD fdwClipPrecision, // clipping precision
    DWORD fdwQuality,     // output quality
    DWORD fdwPitchAndFamily, // pitch and family
    LPCTSTR lpszFace     // pointer to typeface name string
);
```

Optional Parameters: Class

```
class Computer
{
    private String name = "";
    private String motherboard = "";
    private String harddisk = "";
    private String ram = "";
    private String cpu = "";

    public Computer(String name, String motherboard = "",
                    String harddisk = "", String ram = "", String cpu = "")
    {
        this.name = name;
        this.motherboard = motherboard;
        this.harddisk = harddisk;
        this.ram = ram;
        this.cpu = cpu;
    }
}
```

Optional Parameters: Example

```
static void Main(string[] args)
{
    // normal initialisation

    Computer laptop = new Computer("Laptop", "ASUS", "250GB", "2GB",
        "Intel Core 2 Duo");

    // Optional parameters

    Computer junkserver = new Computer("JunkServer", "Gigabyte", "4TB");
}
```



Named Parameters

```
static void Main(string[] args)
{
    // normal initialisation

    Computer laptop = new Computer("Laptop", "ASUS", "250GB", "2GB",
        "Intel Core 2 Duo");

    // Optional parameters

    Computer junkserver = new Computer("JunkServer", "Gigabyte", "4TB");

    // Named parameters

    Computer gamepc = new Computer("GamePC", ram: "8GB",
        cpu: "Intel Core i7");
}
```

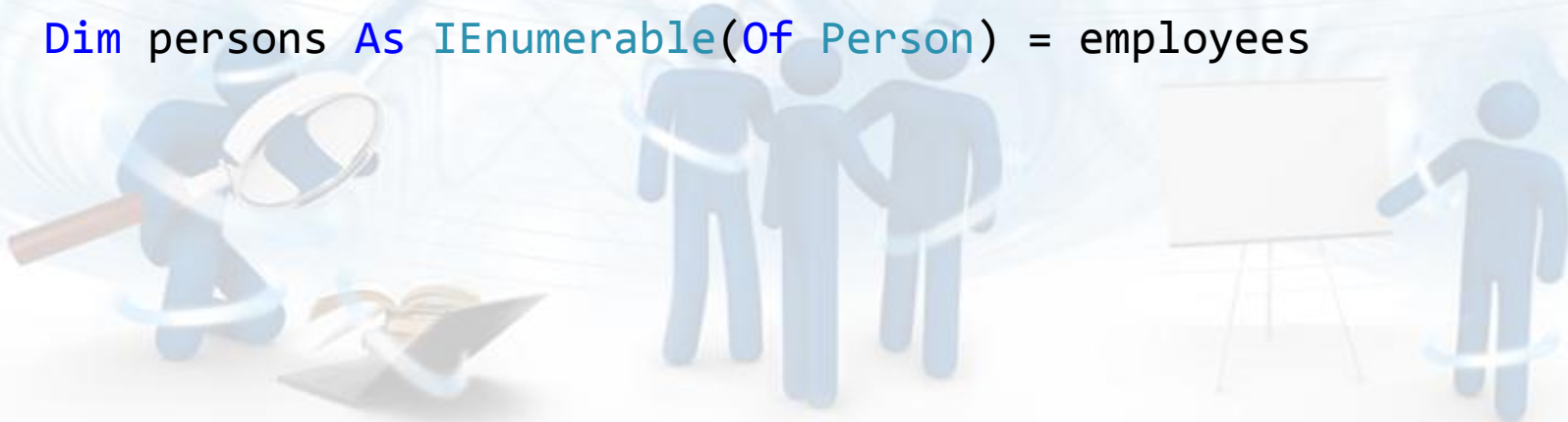
Covariance and Contravariance

- Covariance: derived type in generic
- Contravariance: ancestor in generic
- Applies to:
 - IEnumerable(T) (T is covariant)
 - IEnumerator(T) (T is covariant)
 - IQueryable(T) (T is covariant)
 - IGrouping(TKey, TElement) (TKey and TElement are covariant)
 - IComparer(T) (T is contravariant)
 - IEqualityComparer(T) (T is contravariant)
 - IComparable (T is contravariant)

Covariance Example

- Employee is a Person
- Assign list of employees to list of persons

```
Dim employees As IEnumerable(Of Employee)  
    = New List(Of Employee)  
Dim persons As IEnumerable(Of Person) = employees
```



Office Programming



Office Programming

- Named and optional parameters
- Indexed properties (COM only)
- Auto-dynamic COM objects
- Optional ref keyword
- Dynamic types (duck typing)
 - Also applies to dynamic languages
- Embedded type information



ASP .NET



ASP .NET

- SEO Enhancements
 - MetaKeywords and MetaDescription
 - URL Routing
 - `RouteTable.Routes.MapPageRoute()`
 - `RouteTable.Routes.Add()`
 - `Page.RouteData.Values["querystringfield"]`
 - Permanent Redirect
 - `Response.RedirectPermanent()` – HTTP 301
 - `Response.RedirectToRoute()` – HTTP 302
 - Old Response Redirect => HTTP 302
- Snippets
 - HTML, ASP .NET and JavaScript

New Languages



New Languages

- F#
- Dynamic Language Runtime
 - IronPython
 - IronRuby

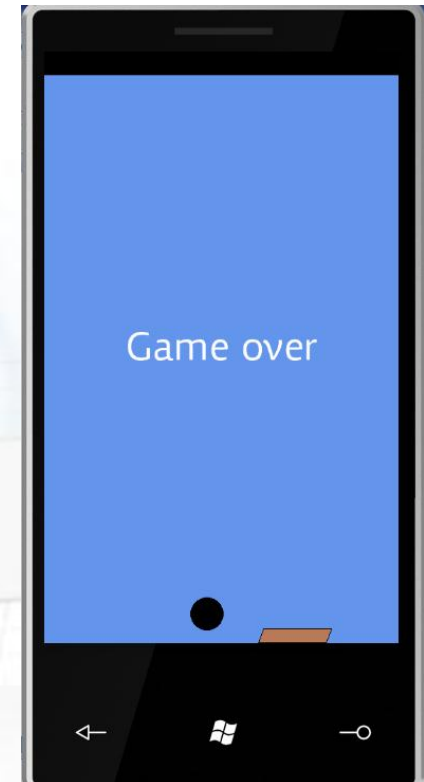


Windows Phone 7



Windows Phone 7

- Windows Phone 7 support in VS2010
- Uses emulator
- Supports
 - Windows Phone 7 Applications
 - Silverlight
 - XNA 4



Wrap up



Summary

- New IDE features
- .NET 4: C# meets VB
- Office Programming
- ASP .NET
- New Languages
- Windows Phone 7

Questions

- Ask away
- Email: dandago@gmail.com

