What’s new in VS 2012 & .NET 4.5

# Setup

* Make sure console font is nice and big
* Copy the ef diagram, hello tech days, WF DEMO, Web Forms, Help pages demos to desktop
* Install Visual Studio 2010 SP 1
* Install Visual Studio 2012 + Update 2
* Start up zoom it
* Set VS to start with IE
* Hide the document outline window
* HELLO.txt file needed
* Clean up Localdb
* Turn off Notifications
* No pin tabs in Chrome
* **DO PRESENTON!**

# Risk less VS

Show the hello techdays demo in VS 2010 & run it – then open in VS 2012 and then back to VS 2010.

Remember to talk about WPF designer improvements

# Culture Demo

Maybe a good time to do Win+4 demo

Cold start VS 2012 – talk amount speed & colours

New console app – demo the format

1. var amount = string.Format("{0:C}",234.123);
2. Console.WriteLine(amount);
3. Console.ReadLine();

Configure the application to use all code analysis settings, Run analyse to generate the issue with it.

<< SLIDE x2 >>

Add in the culture info

1. var amount = string.Format(new CultureInfo("en-us"), "{0:C}",234.123);

Talk about the annoying-ness of having this all the time – solved

1. CultureInfo.DefaultThreadCurrentCulture = new CultureInfo("en-us");

# CallerMemberName Demo

New WPF application and add some WPF

1. <TextBlock Text="{Binding Counter,FallbackValue=0}"/>

Then use the GUI tool to create a style, and then user the property window to adjust the style. Talk about the fact it is the blend UI tooling. Head to code and add:

1. this.DataContext = new Data();
2. /\* elided \*/
3. public class Data : INotifyPropertyChanged
4. {
5. private int counter;
6. public int Counter
7. {
8. get { return counter; }
9. set
10. {
11. counter = value;
12. RaisePropChanged("Counter");
13. }
14. }
15. public Data()
16. {
17. var timer = new DispatcherTimer();
18. timer.Interval = TimeSpan.FromSeconds(0.5);
19. timer.Tick += timer\_Tick;
20. timer.Start();
21. }
22. void timer\_Tick(object sender, EventArgs e)
23. {
24. Counter++;
25. }
26. private void RaisePropChanged(string propertyName)
27. {
28. if (PropertyChanged != null)
29. {
30. PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
31. }
32. }
33. public event PropertyChangedEventHandler PropertyChanged;
34. }

Change the binding to count, and the property but not the magic string – demo and fail.

Now fix it with caller info

1. RaisePropChanged();
2. }
3. }
4. private void RaisePropChanged([CallerMemberName]string propertyName = "")
5. {
6. if (PropertyChanged != null)
7. {
8. PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
9. }
10. }

# Workflow

Open WF demo

Run, show it doesn’t work.

Talk about loop – drag on assign. Talk about C# features.

Add annotation & pin

Search (Ctrl+F) for toilet

Great time to show quick launch – then document outline

Slide about versioning

# Async Demo

New WPF application and setup the XAML

1. <ProgressBar IsIndeterminate="True" HorizontalAlignment="Stretch" Height="50" VerticalAlignment="Top"/>
2. <Button Content="Button" HorizontalAlignment="Left" Margin="74,102,0,0" VerticalAlignment="Top" Width="75" Click="Button\_Click\_1"/>
3. <TextBlock x:Name="resultTextBlock" HorizontalAlignment="Left" Margin="284,93,0,0" TextWrapping="Wrap" Text="TextBlock" VerticalAlignment="Top" FontSize="48"/>

Code behind is

1. private void Button\_Click\_1(object sender, RoutedEventArgs e)
2. {
3. var result = 0;
4. for (int i = 0; i < 300; i++)
5. {
6. Thread.Sleep(10);
7. result += i;
8. }
9. resultTextBlock.Text = result.ToString();
10. }

Demo it and show the freezing – sad panda.

Let’s do the processing in a thread… fuck

1. ThreadPool.QueueUserWorkItem(s =>
2. {
3. var result = 0;
4. for (int i = 0; i < 300; i++)
5. {
6. Thread.Sleep(10);
7. result += i;
8. }
9. resultTextBlock.Text = result.ToString();
10. });

Okay move the bits outside – double fuck

1. var result = 0;
2. ThreadPool.QueueUserWorkItem(s =>
3. {
4. for (int i = 0; i < 300; i++)
5. {
6. Thread.Sleep(10);
7. result += i;
8. }
9. });
10. resultTextBlock.Text = result.ToString();

Add in events & dispatcher

1. public MainWindow()
2. {
3. InitializeComponent();
4. OnCalculated += MainWindow\_OnCalculated;
5. }
6. void MainWindow\_OnCalculated(int obj)
7. {
8. Dispatcher.BeginInvoke(new Action(() =>
9. {
10. resultTextBlock.Text = obj.ToString();
11. }));
12. }
13. public event Action<int> OnCalculated;
14. private void Button\_Click\_1(object sender, RoutedEventArgs e)
15. {
16. ThreadPool.QueueUserWorkItem(s =>
17. {
18. var result = 0;
19. for (int i = 0; i < 300; i++)
20. {
21. Thread.Sleep(10);
22. result += i;
23. }
24. OnCalculated(result);
25. });
26. }

Okay about .NET 4 tasks – bloody messy!

1. Task.Factory.StartNew(() =>
2. {
3. var result = 0;
4. for (int i = 0; i < 300; i++)
5. {
6. Thread.Sleep(10);
7. result += i;
8. }
10. return result;
11. }).ContinueWith(t =>
12. {
13. resultTextBlock.Text = t.Result.ToString();
14. }, TaskScheduler.FromCurrentSynchronizationContext());

Async

1. var result = 0;
2. await Task.Run(() =>
3. {
4. for (int i = 0; i < 300; i++)
5. {
6. Thread.Sleep(10);
7. result += i;
8. }
9. });
11. resultTextBlock.Text = result.ToString();

Let’s see how this will get better with the new Task.Delay timeout

1. var result = 0;
2. var t = Task.Factory.StartNew(() =>
3. {
4. for (int i = 0; i < 300; i++)
5. {
6. Thread.Sleep(10);
7. result += i;
8. }
9. });
10. var r = await Task.WhenAny(t, Task.Delay(1000));
11. if (r == t)
12. {
13. resultTextBlock.Text = result.ToString();
14. }
15. else
16. {
17. resultTextBlock.Text = "time out";
18. }

Cancellation options – add a cancel button

1. private CancellationTokenSource cancelSource;
2. private async void Button\_Click\_1(object sender, RoutedEventArgs e)
3. {
4. this.cancelSource = new CancellationTokenSource();
5. var result = 0;
6. var t = Task.Factory.StartNew(() =>
7. {
8. for (int i = 0; i < 300; i++)
9. {
10. Thread.Sleep(10);
11. result += i;
12. }
13. });
14. var r = await Task.WhenAny(t, Task.Delay(1000, cancelSource.Token));
15. if (r == t)
16. {
17. resultTextBlock.Text = result.ToString();
18. }
19. else
20. {
21. if (r.Status == TaskStatus.Canceled)
22. {
23. resultTextBlock.Text = "cancelled";
24. }
25. else
26. {
27. resultTextBlock.Text = "time out";
28. }
29. }
30. }
31. private void Button\_Click\_2(object sender, RoutedEventArgs e)
32. {
33. cancelSource.Cancel();
34. }

Real world time – lets show the IO/Stream API async – add button & drop in prebuilt “old” way.

Change to 64bit mode

1. var stream = new MemoryStream();
2. using (var fileStream = new FileStream(@"C:\Users\Robert\Desktop\hello.txt", FileMode.Open, FileAccess.Read))
3. {
4. await fileStream.CopyToAsync(stream);
5. }
6. resultTextBlock.Text = stream.Length.ToString();

Part II

# Setup

* Make sure web forms demo is pre-opened
* Warm up app

# EF Demo.

New console app

Nuget EF 5 – double check the pre-release stuff

Add code:

1. public class Person
2. {
3. public int Id { get; set; }
4. public string Name { get; set; }
5. public Gender Gender { get; set; }
6. public int Age { get; set; }
7. }
8. public enum Gender
9. {
10. Male
11. }
12. class MyDB : DbContext
13. {
14. public DbSet<Person> Persons { get; set; }
15. }
16. class Program
17. {
18. static void Main(string[] args)
19. {
20. using (var db = new MyDB())
21. {
22. db.Persons.Add(new Person()
23. {
24. Age = 30,
25. Gender = Gender.Male,
26. Name = "Robert"
27. });
28. db.SaveChanges();
29. }
30. Console.WriteLine("Done");
31. Console.ReadLine();
32. }
33. }

Run it works – show the DB in server explorer! Talk about localdb!

Change person to:

1. public class Person
2. {
3. public int PersonId { get; set; }
4. [Required]
5. public string Name { get; set; }
6. public Gender Gender { get; set; }
7. public DateTime DateOfBirth { get; set; }
8. }

Then Package manager **enable-migrations**, run **add-migration** then **update-database** in package command windows

## Demo the EF diagram app

Show out the box graph

Switch to the Entity model browser

Add new diagram and add entity types from the browser

Then show the coloured original diagram

# Web Dev Demo

Start the web forms app:

* Play the guessing game
* Pretty URLS
* Login
* Responsive UI

New project -> MVC 4 -> Talk about the integration here -> New SPA, run demo and talk about the experience in SPA

Show the NUGET update option here

<< Show Slide about SPA >>

Next switch to chrome and show off the: /api/todolist API

Let’s build one, so new MVC, empty project. Add new EF DB, connect to board games database. **BUILD**. Talk about auto generate, but let’s do empty. Create EF model to board games and then following code for Web API:

public class GamesController : ApiController

{

    public Game GetSingleGame(int id)

    {

        using (var db = new BoardgamesEntities())

        {

            var game = (from g in db.Games

                        where g.Id == id

                        select g).Single();

            return game;

        }

    }

    public IEnumerable<Game> GetTop10Games()

    {

        using (var db = new BoardgamesEntities())

        {

            return (from g in db.Games

                    orderby g.Rank

                    select g).Take(10).ToArray();

        }

    }

}

Demo in chrome

<http://localhost:4690/api/games?id=2>

http://localhost:4690/api/games

talk about discoverability issues. Nuget > web api help page

Run, showr off /help

Load up the futureAPI Help demo, show that for help.

### BEST IDE

New > ASP.NET MVC 4 > Internet Application

Show off page inspector, which lets you show off temp tab

Go into CSS & show the CSS editor – colour editor (RGB as well).

CSS snippet: **Background-origin**

Web essentials css minify.

Cut out content, del file and replace with site.less and do a quick edit of it.

Demo in IE, show network usage

Add to top of bundles: **BundleTable.EnableOptimizations = true;**

Re-run and show again in IE

<< SLIDE >>

Show code behind for CSS, talk about server usage of bundling vs. pre-bundling & that we can add in the bundling.

Search for HeroAccent – show preview image, show browser support then Base64 image then run and profile again

<< SLIDE >>

Do a quick Zen coding demo:

div#main>span.content>ul.menu>li.menuItem\*5>a{click me $$}

Then do tag updating