

The slide features the IT DevCon logo at the top left, followed by the text "European Delphi Conference". Below this is the date "14, 15 november 2013 - VERONA [Italy]". The main title "An introduction to MVVM in Delphi" is centered. To the right, the speaker's name "Jeroen Wiert Pluimers" and website "<http://wiert.me>" are listed. The bottom right corner contains the "bit Time software" logo.

IT DevCon
European Delphi Conference

14, 15 november 2013 - VERONA [Italy]

An introduction to MVVM in
Delphi

Jeroen Wiert Pluimers
<http://wiert.me>

bit Time software



The slide features the IT DevCon logo at the top left, followed by the text "European Delphi Conference". The background is a stylized graphic of overlapping circles in blue, green, and orange. A dark blue horizontal bar across the middle contains the text "The third object". A light blue horizontal bar at the bottom contains the text "Making the UI lighter.".

IT DevCon
European Delphi Conference

The third object

Making the UI lighter.

The 3rd object

IT DevCon European Delphi Conference

- All too often, applications are like these:

or, if you are lucky

- Adding a 3rd object makes sense.
- It all comes down to `Separation of Concerns`:
 - Cutting business logic away from your UI
 - Swapping your UI (VCL, FireMonkey, Mobile, Web, ...)
 - Making it easier to test UI

Pictures: <http://paulstovell.com/blog/third-object>

Martin Fowler: Presentation Model

IT DevCon European Delphi Conference

Sequence diagram illustrating the interaction between the Album Window, Album Presentation Model, and Album objects:

```

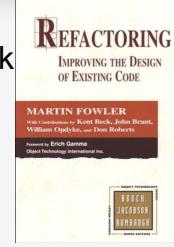
sequenceDiagram
    participant view
    participant pmod as "prod: presentation model"
    participant album
    view->>pmod: classical box clicked
    activate pmod
    pmod->>view: classical_checked
    pmod->>album: pmod.isClassical = view.classical.checked
    view->>album: view.composerField.enabled = pmod.isComposerEnabled
    activate album
    album->>view: apply
    activate view
    view->>album: album.isClassical = presentationModel.isClassical
    deactivate view
    deactivate album
    deactivate pmod
  
```

- 2004; Martin Fowler's article "Presentation Model"
- <http://martinfowler.com/eaaDev/PresentationModel.html>

Martin Fowler: Refactoring Book

IT DevCon European Delphi Conference

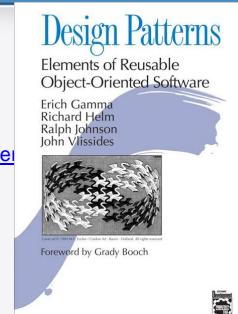
- 1999: Really, really, important *Refactoring* book
 - with a common vocabulary about recipes for improving existing code.
 - http://en.wikipedia.org/wiki/Code_refactoring
 - Refactoring: *Improving the Design of Existing Code*
 - Site:
 - <http://www.refactoring.com/>
 - <http://www.refactoring.com/catalog/>
 - Wiki:
 - <http://c2.com/cgi/wiki?RefactoringImprovingTheDesignOfExistingCode>
 - 1999: Hardcover
 - <http://www.amazon.com/Refactoring-Improving-Design-Existing-Code/dp/0201485672>
 - 2012: Kindle
 - <http://www.amazon.com/Refactoring-Improving-Existing-Ad-Wesley-Technology-ebook/dp/B007WTFWJ6>
 - Martin Fowler
 - http://en.wikipedia.org/wiki/Martin_Fowler




MVVM is about Patterns

IT DevCon European Delphi Conference

- 2004: The reference on Patterns
 - with a common vocabulary about recipes for developing new code.
 - [http://en.wikipedia.org/wiki/Design_pattern_\(computer_science\)](http://en.wikipedia.org/wiki/Design_pattern_(computer_science))
 - Design Patterns: “Elements of Reusable Object-Oriented Software”
 - http://en.wikipedia.org/wiki/Design_Patterns
 - Written by “the Gang of Four”:
 - <http://c2.com/cgi/wiki?GangOfFour>
 - Erich Gamma
 - Richard Helm
 - Ralph Johnson
 - John Vlissides
 - Martin Fowler on it: <http://martinfowler.com/bliki/GangOfFour.html>
 - In my view the Gang of Four is the best book ever written on object-oriented design - possibly of any style of design. This book has been enormously influential
 - The 3rd object is “just” a (relatively) new way of using patterns.



3rd object: adds code

- The term "**View model**" is a major cause of confusion in understanding the pattern when compared to the more widely implemented [MVC](#) or [MVP](#) patterns.
- The role of the controller or presenter of the other patterns has been substituted with the framework binder (e.g., [XAML](#)) and view model as mediator and/or converter of the model to the binder.
 - http://en.wikipedia.org/wiki/Model_View_Model#cite_ref-10

MVC, MVP, MVVM

- MVC is like a circle
- MVP is multi-way
- MVVM is like MVP,
 - with automated M-V and V-VM bindings to allow real UI designers to work on the V
 - VM like P is close to the V, but testable
- Further reading
 - http://en.wikipedia.org/wiki/Model_View_Model
 - <http://joel.inpointform.net/software-development/mvvm-vs-mvp-vs-mvc-the-differences-explained/>
 - <http://nirajrules.wordpress.com/2009/07/18/mvc-vs-mvp-vs-mvvm/>
 - <http://alexander.lsd.lg.ua/2010/05/mvvm-model-view-view-model-design-pattern-for-net-windows-forms-winforms/>
 - <http://blogs.msdn.com/b/johngossman/archive/2005/10/08/478683.aspx>

Platforms with MVVM

```

classDiagram
    class ViewModelBase {
        <<ViewModel base class>>
        +INotifyPropertyChanged
    }
    class View
    class ViewModel
    class Model
    interface INotifyPropertyChanged
    interface INotifyDataErrorInfo

    View --> ViewModel
    ViewModel --> Model
    ViewModel "Inherits" --> ViewModelBase
    ViewModel <|-- INotifyPropertyChanged
    ViewModel <|-- INotifyDataErrorInfo
  
```

- .NET
 - Caliburn <http://caliburn.codeplex.com/> <http://caliburnmicro.codeplex.com>
 - PRISM <http://msdn.microsoft.com/en-us/library/gg430869>
 - MvvmCross <https://github.com/MvvmCross>
- JavaScript:
 - KnockoutJS <http://knockoutjs.com>
 - AngularJS <http://angularjs.org>
- Java:
 - AndroidBinding <https://github.com/queei/AndroidBinding>
 - ZK <http://www.zkoss.org>
 - Bambi <https://github.com/S73417H/bambi-mvvm>
- Cocoa: the View Controller in the [Cocoa Design Patterns](#)

MVVM goals

- Write as little code as possible by using
 - "Program to an 'interface', not an 'implementation'." • (Gang of Four 1995:18)
 - "Favor 'object composition' over 'class inheritance'." • (Gang of Four 1995:20)
 - A structured 3-layer approach
- The `Caliburn for Delphi` framework
 - allows the examples of this session to work
 - implements MVVM, and a lot more
 - is very similar to the C# Caliburn / Caliburn.Micro frameworks <http://channel9.msdn.com/Events/MIX/MIX10/EX15> <https://github.com/bryanhunter/CaliburnMicroTalk>
 - is written by Stefan Glienke (of DSharp) <http://stackoverflow.com/users/587106/stefan-glienke> and enhanced by Marko Vončina <http://www.linkedin.com/pub/marko-voncina/85/741/895>

The start of Caliburn Micro

IT DevCon
European Delphi Conference

F8X 10
THE NEXT WEB NOW
March 15-17th, 2010, Las Vegas

Build Your Own
MVVM Framework

Rob Eisenberg
Blue Spire
rob@bluespire.com
@EisenbergEffect

0:00:51

Caliburn for Delphi features and fundamentals

- Features:
 - MVVM (Model-View-ViewModel)
 - Convention over Configuration
 - Event Aggregator (publish-subscribe)
 - Coroutines
 - Conductors and Screens
 - Bootstrapper
- Fundamentals
 - Interfaces
 - Attributes
 - Generics
- We will cover some of these here, starting with MVVM.

IT DevCon
European Delphi Conference

Global structure

IT DevCon European Delphi Conference

- View
 - What the user sees
 - Are dumb
- Model
 - The data
- ViewModel
 - Glue between View and Model:
 - actions/bindings of the View
 - bindings on the model
 - contains most code
- Model and ViewModel
 - are testable

```

graph LR
    View[View] -- "UI events" --> ViewModel[ViewModel]
    ViewModel -- "PropertyChanged events" --> View
    ViewModel <--> Model[Model]
    Model -- "Model change events" --> ViewModel
    Model -- "Update" --> ViewModel
    Model -- "Read" --> ViewModel
    ViewModel -- "ViewModel data" --> View
  
```

<http://stackoverflow.com/questions/5421874/basic-concepts-of-mvvm-what-should-a-viewmodel-do>

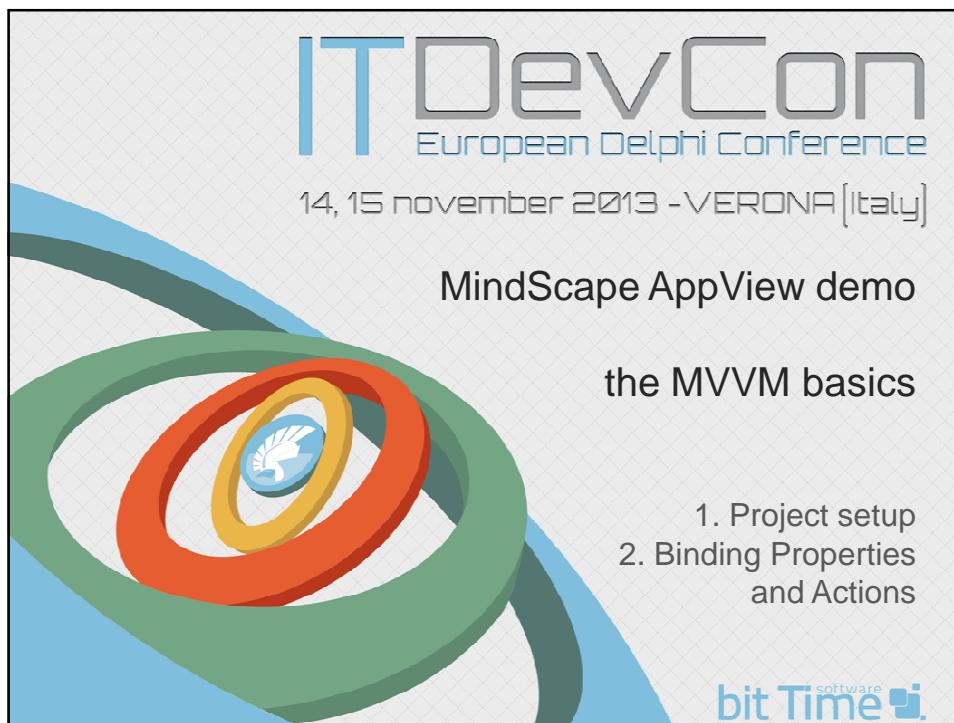
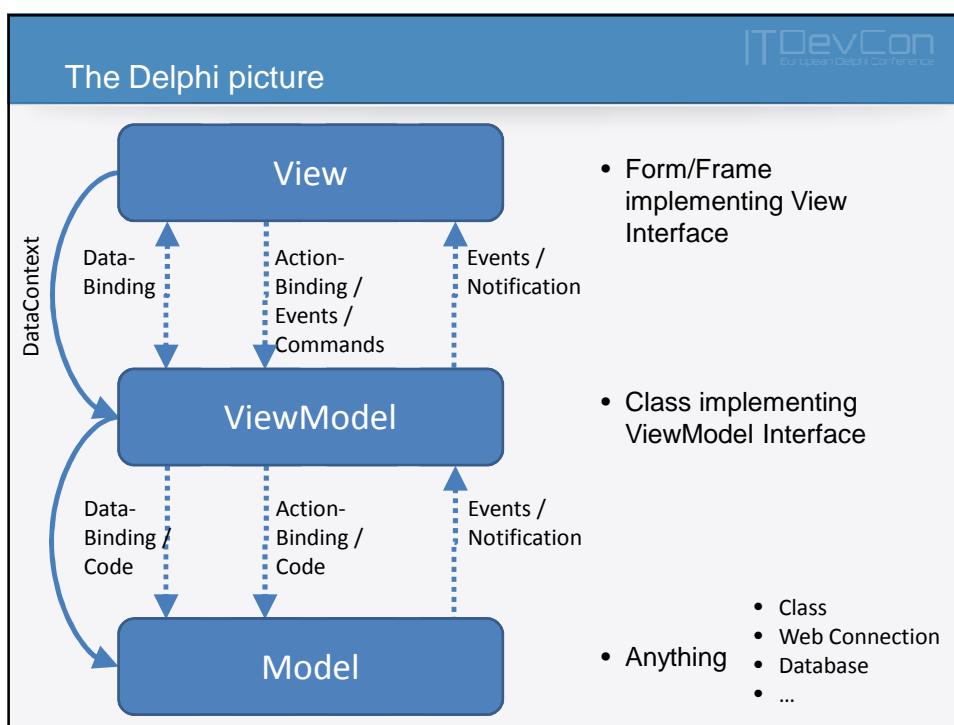
Global structure

IT DevCon European Delphi Conference

- Everything in your application starts with the View/ViewModel combo
- Even the main Window is a View (and has a ViewModel)
 - Those are usually named either of these
 - ShellView/ShellViewModel
 - AppView / AppViewModel
- It is up to the ViewModel how to do the Model
- So we will concentrate on the View and ViewModel:

```

graph LR
    View[View] -- "UI events" --> ViewModel[ViewModel]
    ViewModel -- "PropertyChanged events" --> View
    ViewModel <--> Model[Model]
    Model -- "Model change events" --> ViewModel
    Model -- "Update" --> ViewModel
    Model -- "Read" --> ViewModel
    ViewModel -- "ViewModel data" --> View
  
```





MindScape AppView
Part A: getting started

<http://www.mindscapehq.com/blog/index.php/2012/01/12/caliburn-micro-part-1-getting-started/>

Example: MindScape AppView – part A; 1;2;3/...

IT DevCon
European Delphi Conference

1. Create VCL project with these units and forms:
 1. AppViewForm
 - (VCL Form `TAppView` with name `AppView`)
 2. ApplInterfaces
 - (interface `IAppViewModel`)
 3. AppViewModel
 - (class `TAppViewModel` implementing `IAppViewModel`)
2. Change these paths:
 1. Either in the Project Search Path (which I like most)
 2. Or in the Delphi Library Path:
 - ..\Source\PresentationModel;..\Source\Validation;..\External\DSsharp\Aspects;..\External\DSsharp\Bindings;..\External\DSsharp\Collections;..\External\DSsharp\ComponentModel;..\External\DSsharp\Core;..\External\DSsharp\DelphiWebScript;..\External\DSsharp\DevExpress;..\External\DSsharp\Interception;..\External\DSsharp\Logging;..\External\DSsharp\Testing;..\External\DSsharp\Windows;\$(Spring)\Source\Base;\$(Spring)\Source\Base\Collections;\$(Spring)\Source\Base\Reflection;\$(Spring)\Source\Core\Container;\$(Spring)\Source\Core\Services
3. Add conditional defines
 1. To DEBUG
 - CodeSite;DEBUG

Example: MindScape AppView – Part A; 4;5/...

4. Chang AppView unit to have
- an 'initialization' section and
 - no more `var AppView: TAppView`

like this:

```
unit AppViewForm;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms, Dialogs, DSharp.Bindings.VCLControls;
type
  TAppView = class(TForm)
  end;
implementation
{$R *.dfm}
initialization
  TAppView.ClassName;
end.
```

5. Change AppViewModel unit to

- An initialization section

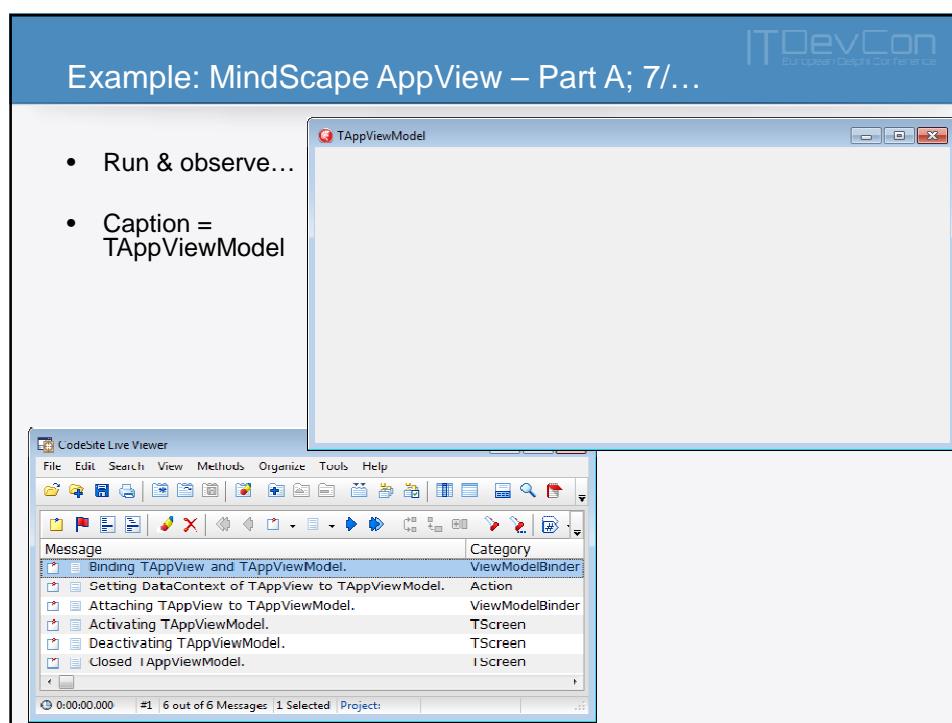
```
unit AppViewModel;
interface
uses
  AppInterfaces,
  DSharp.PresentationModel;
type
  TAppViewModel = class(TScreen, IAppViewModel)
  end;
implementation
initialization
  TAppViewModel.ClassName;
end.
```

Example: MindScape AppView – Part A; 6/...

6. Change main program to

- Include DSharp units
- Start using the IAppViewModel or TAppViewModel

```
program MindScape_AppViewVCL;
uses
{$ifdef CodeSite}
  DSharp.Logging.CodeSite,
{$endif CodeSite}
  DSharp.PresentationModel.VCLApplication,
  Forms,
  AppViewForm in 'AppViewForm.pas' {AppView},
  AppViewModel in 'AppViewModel.pas',
  AppInterfaces in 'AppInterfaces.pas';
{$R *.res}
begin
  Application.Initialize;
  Application.MainFormOnTaskbar := True;
  ReportMemoryLeaksOnShutdown := True;
{$ifdef DEBUG}
  Application.WithDebugLogger();
{$endif DEBUG}
{$ifdef CodeSite}
  Application.WithLogger<TCodeSiteLog>;
{$endif CodeSite}
  Application.Start<IAppViewModel>();
//  Application.Start<TAppViewModel>();
end.
```



Example: MindScape AppView – part B; 1/...

1. Modify the AppViewModel

- Add a Count property, with FCount backing field and SetCount method having a NotifyOfPropertyChange call:

```
unit AppViewModel;
...
type
  TAppViewModel = class(TScreen, IAppViewModel)
    strict private
      FCount: Integer;
    strict protected
      procedure SetCount(const Value: Integer); virtual;
    public
      property Count: Integer read FCount write SetCount;
    end;
...
procedure TAppViewModel.SetCount(const Value: Integer);
begin
  if FCount <> Value then
  begin
    FCount := Value;
    NotifyOfPropertyChange('Count');
  end;
end;
...
```

Example: MindScape AppView – part B; 2/...

2. Modify the AppView

- Add a TEdit control with name Count:

```
unit AppViewForm;
...
  DSharp.Bindings.VCLControls, StdCtrls;
type
  TAppView = class(TForm)
    Count: TEdit;
  end;
...
```

Example: MindScape AppView – part B; 3/...

IT DevCon
European Delphi Conference

3. Run

- It fails to show the content of the Count property!
- The Count element (TEdit component) could not be bound to the ViewModel.

CodeSite Live Viewer

File Edit Search View Methods Organize Tools Help

Message

Binding TAppView and TAppViewModel.
Setting DataContext of TAppView to TAppViewModel.
Attaching TAppView to TAppViewModel.
Action Convention Not Applied: No actionable element for procedure AppViewModel.TAppViewModel.SetCount(const Value: string).
Binding Convention Not Applied: No conventions configured for Count(Vcl.StdCtrls.TEdit)...
Activating TAppViewModel.
Deactivating TAppViewModel.
Closed TAppViewModel.

Binding Convention Not Applied: No conventions configured for Count(Vcl.StdCtrls.TEdit)
of AppView(AppViewForm.TAppView)
of @\\$03925AE0(Vcl.Forms.TApplication).

0:00:00.003 #5 8 out of 8 Messages 1 Selected Project:

Example: MindScape AppView – part B; 4;5/...

IT DevCon
European Delphi Conference

4. Modify the AppView

- Make sure that `DSharp.Bindings.VCLControls` is at the end of the uses list:

```
Controls, Forms, Dialogs, StdCtrls,  
DSharp.Bindings.VCLControls;
```

5. Run

- The unit `DSharp.Bindings.VCLControls` manages the binding

CodeSite Live Viewer

File Edit Search View Methods Organize Tools Help

Message

Binding Convention Applied: AppViewModel.TAppViewModel.Count to property Text of Element Count(DSharp.Bindings.VCLControls.TEdit)
Activating TAppViewModel.
Deactivating TAppViewModel.
Closed TAppViewModel.

Binding Convention Applied: AppViewModel.TAppViewModel.Count to property Text of Element Count(DSharp.Bindings.VCLControls.TEdit)
of AppView(AppViewForm.TAppView)
of @\\$02125AE0(Vcl.Forms.TApplication).

0:00:00.003 #5 8 out of 8 Messages 1 Selected Project:

Example: MindScape AppView – part B; 6/...

6. Modify the AppViewModel
- Add two consts, two functions and two methods for Incrementing/Decrementing Count:
- ```

unit AppViewModel;
...
type
 TAppViewModel = class(TScreen, IAppViewModel)
 ...
 public
 const MinimumCount = -10;
 const MaximumCount = +10;
 function CanDecrementCount(): Boolean; virtual;
 function CanIncrementCount(): Boolean; virtual;
 procedure DecrementCount(); virtual;
 procedure IncrementCount(); virtual;
 end;
...
function TAppViewModel.CanDecrementCount: Boolean;
begin
 Result := Count > MinimumCount;
end;
function TAppViewModel.CanIncrementCount: Boolean;
begin
 Result := Count < MaximumCount;
end;
procedure TAppViewModel.DecrementCount;
begin
 Count := Count - 1;
end;
procedure TAppViewModel.IncrementCount;
begin
 Count := Count + 1;
end;
...
end.
```

## Example: MindScape AppView – part B; 7;8/...

### 7. Modify the AppView

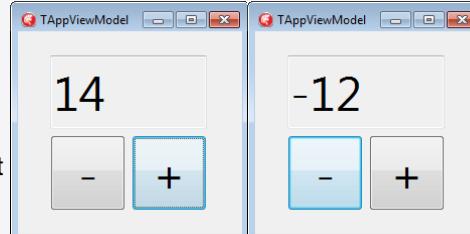
- Add a two TButtons and name them like this:

```

unit AppViewForm;
...
type
 TAppView = class(TForm)
 Count: TEdit;
 IncrementCount: TButton;
 DecrementCount: TButton;
 end;
...
end.
```

### 8. Run

- Observe you can increment and decrement too far



Example: MindScape AppView – part B; 9;10/...

IT DevCon  
European Delphi Conference

9. Observe the CodeSite log:

- ① Action Convention Not Applied: No actionable element for function AppViewModel.TAppViewModel.CanDecrementCount: Boolean.
- ② Action Convention Not Applied: No actionable element for function AppViewModel.TAppViewModel.CanIncrementCount: Boolean.

10. Modify the AppViewModel:

- Refactor the CanDecrementCount/CanIncrementCount functions into properties; change the implementation of SetCount:

```

unit AppViewModel;
...
strict protected
 function GetCanDecrementCount(): Boolean; virtual;
 function GetCanIncrementCount(): Boolean; virtual;
...
public
...
 property CanDecrementCount: Boolean read GetCanDecrementCount;
 property CanIncrementCount: Boolean read GetCanIncrementCount;
end;
...
function TAppViewModel.GetCanDecrementCount(): Boolean;
begin
 Result := Count > MinimumCount;
end;
function TAppViewModel.GetCanIncrementCount(): Boolean;
begin
 Result := Count < MaximumCount;
end;
procedure TAppViewModel.SetCount(const Value: Integer);
begin
 if FCount <> Value then
 begin
 FCount := Value;
 NotifyOfPropertyChange('Count');
 NotifyOfPropertyChange('CanDecrementCount');
 NotifyOfPropertyChange('CanIncrementCount');
 end;
end;
...
end.

```

Example: MindScape AppView – part B; 11/...

IT DevCon  
European Delphi Conference

### 11. Run and observe

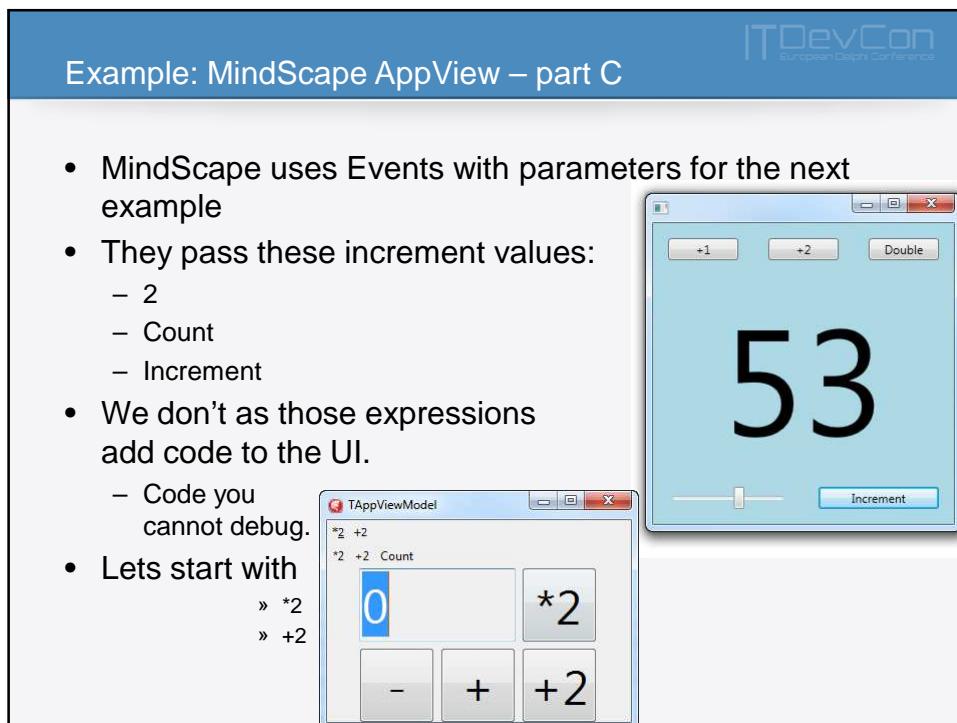
- ① Action Convention Not Applied: No actionable element for function AppViewModel.TAppViewModel.GetCanDecrementCount: Boolean.
- ② Action Convention Not Applied: No actionable element for function AppViewModel.TAppViewModel.GetCanIncrementCount: Boolean.
- ③ Action Convention Not Applied: No actionable element for procedure AppViewModel.TAppViewModel.SetCount(const Value: Integer).
- ④ Can\_\_\_ Binding Convention Applied: AppViewModel.TAppViewModel.CanDecrementCount to property Enabled of Element DecrementCount(DSharp.Bindings.VCLControls.TEdit).
- ⑤ Action Convention Applied: Action procedure AppViewModel.TAppViewModel.DecrementCount on element DecrementCount(DSharp.Bindings.VCLControls.TEdit).
- ⑥ Can\_\_\_ Binding Convention Applied: AppViewModel.TAppViewModel.CanIncrementCount to property Enabled of Element IncrementCount(DSharp.Bindings.VCLControls.TEdit).
- ⑦ Action Convention Applied: Action procedure AppViewModel.TAppViewModel.IncrementCount on element IncrementCount(DSharp.Bindings.VCLControls.TEdit).
- ⑧ Binding Convention Applied: AppViewModel.TAppViewModel.Count to property Text of Element Count(DSharp.Bindings.VCLControls.TEdit).
- ⑨ Activating TAppViewModel.
- ⑩ Deactivating TAppViewModel.
- ⑪ Closed TAppViewModel.

1 Can\_\_\_ Binding Convention Applied: AppViewModel.TAppViewModel.CanIncrementCount to property Enabled of Element IncrementCount(DSharp.Bindings.VCLControls.TEdit) of AppView(AppViewForm.TAppView) of @\\$03A95AE0(Vcl.Forms.TApplication).

The screenshots show three states of the application window titled 'TAppViewModel'. In the first state, the count is '-10' and both the minus and plus buttons are grayed out. In the second state, the count is '-9' and the minus button is grayed out while the plus button is highlighted with a blue border. In the third state, the count is '10' and the plus button is grayed out while the minus button is highlighted with a blue border.



The slide features the IT DevCon logo at the top right. Below it is a decorative graphic of overlapping circles in blue, green, and red. The main title "MindScape AppView" is displayed in white on a dark blue background. Below the title, the subtitle "Part C: events but no action parameters" is shown in white. At the bottom left, there is a link: <http://www.mindscapehq.com/blog/index.php/2012/01/24/caliburn-micro-part-3-more-about-events-and-parameters/>.



The slide title is "Example: MindScape AppView – part C". It contains a bulleted list of points about MindScape's use of events with parameters:

- MindScape uses Events with parameters for the next example
- They pass these increment values:
  - 2
  - Count
  - Increment
- We don't use those expressions add code to the UI.
  - Code you cannot debug.
- Lets start with
  - » \*2
  - » +2

On the right side of the slide, there are two windows. The top window is a desktop application titled "TAppViewModel" showing a large number "53" and a slider labeled "Increment". The bottom window is a screenshot of a user interface with a numeric keypad and buttons for "+1", "+2", and "Double".

## Example: MindScape AppView – part C; 1/...

```

TAppViewModel = class(TScreen, IAppViewModel)
strict protected
 function GetCanIncrementCountBy2(): Boolean; virtual;
 function GetCanMultiplyCountBy2(): Boolean; virtual;
public
 procedure IncrementCountBy2(); virtual;
 procedure MultiplyCountBy2(); virtual;
 property CanIncrementCountBy2: Boolean read GetCanIncrementCountBy2;
 property CanMultiplyCountBy2: Boolean read GetCanMultiplyCountBy2;
end;
...
function TAppViewModel.GetCanIncrementCountBy2(): Boolean;
begin
 Result := Count + 1 < MaximumCount;
end;
function TAppViewModel.GetCanMultiplyCountBy2(): Boolean;
begin
 Result := (Count * 2 <= MaximumCount) and (Count * 2 >= MinimumCount);
end;
procedure TAppViewModel.IncrementCountBy2();
begin
 Count := Count + 2;
end;
procedure TAppViewModel.MultiplyCountBy2();
begin
 Count := Count * 2;
end;
procedure TAppViewModel.SetCount(const Value: Integer);
begin
 ... NotifyOfPropertyChange('CanIncrementCountBy2');
 ... NotifyOfPropertyChange('CanMultiplyCountBy2'); ...
end;

```

## Example: MindScape AppView – part C; 2/...

2. Modify the AppView

- Add ActionManager, Actions + bindings, ActionToolBar, ActionMenuBar, Buttons, Menultems, uses list:

```

unit AppViewForm;
...
uses
 ... Vcl.ActnMan, Vcl.ActnCtrls, System.Actions,
 Vcl.ActnList, Vcl.PlatformDefaultStyleActnCtrls,
 DSharp.PresentationModel,
 DSharp.Bindings.VCLControls;
type
 TAppView = class(TForm)
 ...
 ActionManager1: TActionManager;
 [Binding('OnExecute', '{Binding IncrementCountBy2}')]

 [Binding('Enabled', '{Binding CanIncrementCountBy2}')]

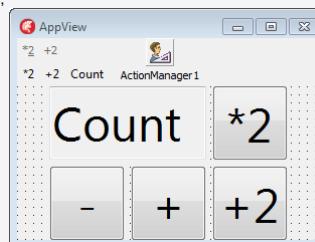
 IncrementByTwo: TAction;

 [Binding('OnExecute', '{Binding MultiplyCountBy2}')]

 [Binding('Enabled', '{Binding CanMultiplyCountBy2}')]

 MultiplyByTwo: TAction;
 ActionToolBar1: TActionToolBar;
 ActionMainMenuBar1: TActionMainMenuBar;
 Button1: TButton;
 Button2: TButton;
 end;
 ...
end.

```



## Example: MindScape AppView – part C; 3/...

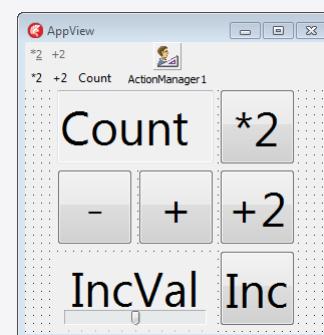
### 3. Common mistakes

- Compiler errors
  - [dcc32 Warning] AppViewForm.pas(19): W1025 Unsupported language feature: 'custom attribute'
  - Solution: In the View, add the unit DSharp.PresentationModel to the uses list.
- Exception:
  - Problem: Project MindScape\_AppViewVCL.exe raised exception
    - class EAssertionFailed with message 'Source is not assigned! (C:\Users\developer\Versioned\Caliburn4D\Source\PresentationModel\DSsharp.PresentationModel.InitializeComponent.pas, line 302)'.
  - Solution: issue in **unit DSsharp.PresentationModel.VCLFramework**
    - Method TPresentationFramework.DoGetParent could not find a Parent (this happened for TActions and is resolved now).
- Explicit bindings have no effect (but give no errors or warnings):
  - Problem: typing errors in the Binding Expression, giving no match for:
    - ViewModel
    - Properties on the ViewModel
    - Methods on the ViewModel
    - Paths in sub-Components on the ViewModel
  - Solution:
    - Fix the typing errors
    - Have the Caliburn team investigate into logging expression issues.

## Example: MindScape AppView – part C; 4/...

### 4. Before adding the “Increment” with “TrackBar” functionality

- Lets observe a few things:
  - The UI has no code, but bindings
    - Automatic bindings
    - Manual bindings through attributes
  - Bindings can be from
    - Events to Methods
    - Properties to Properties
  - For the “Increment” functionality, we want to hold those observations.



## Example: MindScape AppView – part C; 5/...

```
5. Modify the unit AppViewModel;

TAppViewModel = class(TScreen, IAppViewModel)
strict private
 FIncrementValue: Integer;
strict protected
 function GetCanIncrementCountByIncrementValue (): Boolean; virtual;
 procedure SetIncrementValue(const Value: Integer); virtual;
public
 procedure IncrementCountByIncrementValue(); virtual;
 property CanIncrementCountByIncrementValue: Boolean read GetCanIncrementCountByIncrementValue;
 property IncrementValue: Integer read FIncrementValue write SetIncrementValue;
end;
...
function TAppViewModel.GetCanIncrementCountByIncrementValue(): Boolean;
begin
 Result := (Count + IncrementValue >= MinimumCount) and
 (Count + IncrementValue <= MaximumCount);
end;
procedure TAppViewModel.IncrementCountByIncrementValue();
begin
 Count := Count + IncrementValue;
end;
procedure TAppViewModel.SetCount(const Value: Integer);
begin
...
 NotifyOfPropertyChanged('CanIncrementWithValue');
end;
procedure TAppViewModel.SetIncrementValue(const Value: Integer);
begin
 if FIncrementValue <> Value then
 begin
 FIncrementValue := Value;
 NotifyOfPropertyChanged('IncrementValue');
 NotifyOfPropertyChanged('CanIncrementWithValue');
 end;
end;
end;
```

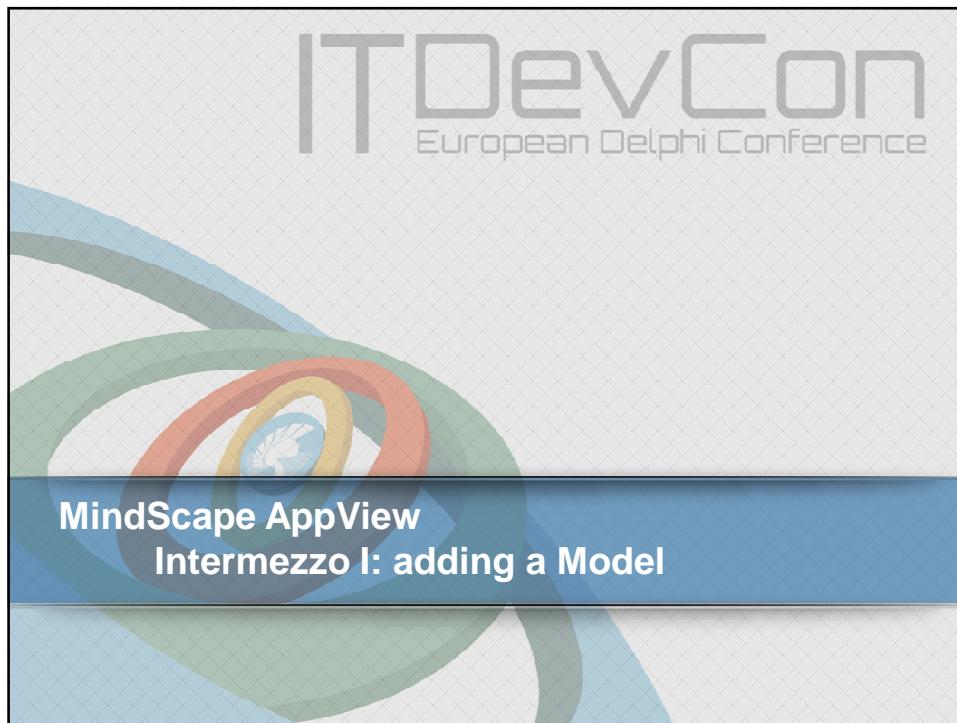
## Example: MindScape AppView – part C; 6/...

### 6. Modify the AppView

- Add a TTrackBar, TLabel and TButton:

```
unit AppViewForm;
...

TAppView = class(TForm)
 IncrementValue: TTrackBar;
 [Binding('Caption', '{Binding IncrementValue}')]
 IncVal: TLabel;
 IncrementCountByIncrementValue: TButton;
end;
...
end.
```



Example: MindScape AppView – Intermezzo I; 1/...

1. The Model could be anything, but since we use Caliburn...

```
unit AppInterfaces;
interface
...
const
 MinimumCount = -10;
 MaximumCount = +10;
type
 [InheritedExport]
 IAppModel = interface
 ['{DD3AABF1-140F-4F78-85E3-2E332218F8AE}']
 function GetCount(): Integer;
 function GetIncrementValue(): Integer;
 procedure SetCount(const Value: Integer);
 procedure SetIncrementValue(const Value: Integer);
 property Count: Integer read GetCount write SetCount;
 property IncrementValue: Integer read GetIncrementValue
 write SetIncrementValue;
 end;
...
end.
```

## Example: MindScape AppView – Intermezzo I; 2/...

2. Add the AppModel unit interface

```

unit AppModel;
interface
uses
 AppInterfaces, IniFiles;
type
 TAppModel = class(TInterfacedObject, IAppModel)
 strict private
 FCount: Integer;
 FIncrementValue: Integer;
 const
 SAppModel = 'AppModel';
 SCount = 'Count';
 SIncrementValue = 'IncrementValue';
 protected
 function CreateIniFile(): TIniFile; virtual;
 function GetCount(): Integer; virtual;
 function GetIncrementValue(): Integer; virtual;
 function GetIniFileName(): string; virtual;
 procedure SetCount(const Value: Integer); virtual;
 procedure SetIncrementValue(const Value: Integer); virtual;
 public
 constructor Create();
 destructor Destroy(); override;
 property Count: Integer read GetCount write SetCount;
 property IncrementValue: Integer read GetIncrementValue write SetIncrementValue;
 property IniFileName: string read GetIniFileName;
 end;

```

## Example: MindScape AppView – Intermezzo I; 3/...

3. Add the AppModel implementation part 1: INI file persistence.

```

...
uses
 SysUtils;
constructor TAppModel.Create();
var
 IniFile: TIniFile;
begin
 inherited;
 IniFile := CreateIniFile();
 try
 Count := IniFile.ReadInteger(SAppModel, SCount, 0);
 IncrementValue := IniFile.ReadInteger(SAppModel, SIncrementValue, 0);
 finally
 IniFile.Free();
 end;
end;
destructor TAppModel.Destroy();
var
 IniFile: TIniFile;
begin
 IniFile := CreateIniFile();
 try
 IniFile.WriteInteger(SAppModel, SCount, Count);
 IniFile.WriteInteger(SAppModel, SIncrementValue, IncrementValue);
 finally
 IniFile.Free();
 end;
 inherited;
end;
function TAppModel.CreateIniFile(): TIniFile;
begin
 Result := TIniFile.Create(IniFileName);
end;

```

## Example: MindScape AppView – Intermezzo I; 4/...

4. Add the AppModel implementation part 2: property logic

```

...
function TAppModel.GetCount(): Integer;
begin
 Result := FCount;
end;
function TAppModel.GetIncrementValue(): Integer;
begin
 Result := FIncrementValue;
end;
function TAppModel.GetIniFileName(): string;
begin
 Result := ChangeFileExt(ParamStr(0), '.ini');
end;
procedure TAppModel.SetCount(const Value: Integer);
begin
 if FCount <> Value then
 begin
 if (Value < MinimumCount) or (Value > MaximumCount) then
 raise ERangeError.CreateFmt('Count value %d out of range %d..%d', [Value,
 MinimumCount, MaximumCount]);
 FCount := Value;
 end;
end;
procedure TAppModel.SetIncrementValue(const Value: Integer);
begin
 FIncrementValue := Value;
end;
end.
```

## Example: MindScape AppView – Intermezzo I; 5/...

5. Modify the AppViewModel to use the AppModel

- Replace all Count/IncrementValue with AppModel.Count/AppModel.IncrementValue:

```

uses
 AppInterfaces, ...
type
 TAppViewModel = class(TScreen, IAppViewModel)
 strict private
 FAppModel: IAppModel;
 strict protected
 property AppModel: IAppModel read FAppModel;
 public
 constructor Create(const AAppModel: IAppModel);
 end;
 ...
constructor TAppViewModel.Create(const AAppModel: IAppModel);
begin
 inherited Create();
 FAppModel := AAppModel;
end;
function TAppViewModel.GetCount(): Integer;
begin
 Result := AppModel.Count;
end;
function TAppViewModel.GetIncrementValue(): Integer;
begin
 Result := AppModel.IncrementValue;
end;
procedure TAppViewModel.SetCount(const Value: Integer);
begin
 ... AppModel.Count := Value;
 ...
end;
procedure TAppViewModel.SetIncrementValue(const Value: Integer);
begin
 ... AppModel.IncrementValue := Value;
 ...
end;
```

## Example: MindScape AppView – Intermezzo I; 6;7/...

### 6. Run

- Boom!

```
Project MindScape_AppViewVCL.exe raised exception
 class $C0000005 with message 'access violation at
 0x00ac607a: read of address 0x00000000'.
 function TAppViewModel.GetCount(): Integer;
 begin
 Result := AppModel.Count;
 end;
```

### 7. You try to use AppModel, but it is nil as the new Constructor did not get called

- Reason 1: forgetting the [InheritedExport]

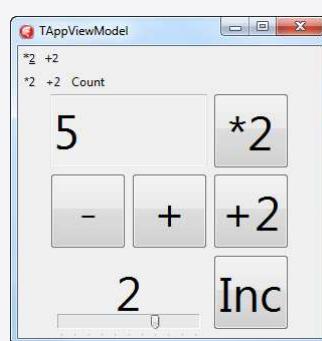
```
type
 [InheritedExport]
 IAppModel = interface
 ['{DD3AABF1-140F-4F78-85E3-2E332218F8AE}']
 unit AppModel;
 ...
 initialization
 TAppModel.ClassName;
 end.
```

## Example: MindScape AppView – Intermezzo I; 8/...

### 8. Run

- It works!

1. It loads initial values like before
2. Each time you run again, Count/IncrementValue are restored to what they were



**IT DevCon European Delphi Conference**

**MindScape AppView  
Intermezzo II: unit testing**

[http://en.wikipedia.org/wiki/Extreme\\_programming](http://en.wikipedia.org/wiki/Extreme_programming)

**Example: MindScape AppView – Intermezzo II; 1/...**

IT DevCon European Delphi Conference

1. Add a Unit Test project
2. Add the AppModel to it
3. Add the AppViewModel to it
4. Generate TTestCase for AppModel
5. Generate TTestCase for AppViewModel
6. Be creative in your test writing

Example: MindScape AppView – Intermezzo II; 2/...

IT DevCon European Delphi Conference

- Run the tests, rinse, repeat...

| Tests | Run | Failures | Errors | Overrides | Test Time   | Total Time   |
|-------|-----|----------|--------|-----------|-------------|--------------|
| 30    | 30  | 0        | 20     | 0         | 0:00:00.008 | 0:00:00.1... |

Test Name Failure Type Message Location  
 ■ TestDecrementC... ERangeError Count value -11 out of range -10..10 \$0075DFEF  
 ■ TestDecrementC... ERangeError Count value -11 out of range -10..10 \$0075DFEF  
 ■ TestDecrementC... ERangeError Count value 11 out of range -10..10 \$0075DFEF

**TestDecrementCount\_Minus10: ERangeError**  
 at \$0075DFEF  
 Count value -11 out of range -10..10

IT DevCon  
European Delphi Conference

14, 15 november 2013 - VERONA [Italy]

MindScape Color demo

beyond the MVVM basics

1. Events

bit Time software



**MindScape Color**  
Part A: getting started

<http://www.mindscapehq.com/blog/index.php/2012/02/01/caliburn-micro-part-4-the-event-aggregator/>

Demo: Color

IT DevCon  
European Delphi Conference

- Color has:
  - Two Models
    - ShellViewModel (comparable to AppViewModel in the first demo)
    - ColorViewModel (handles selection and display of colors)
  - Two Views
    - ShellView (comparable to AppView in the first demo)
    - ColorView (new view showing a color)
  - Displays ColorView inside ShellView
  - Uses Events to communicate

## Example: MindScape Color – part C; 1..3/...

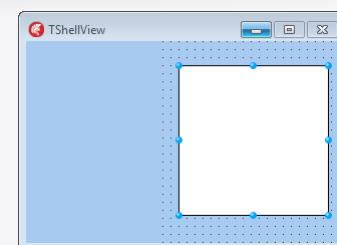
1. Start like AppView, but use ShellView/ShellViewModel instead of AppView/AppViewModel
2. Add a ColorModel/ColorViewModel
3. Make sure you have these interfaces

```
unit Interfaces;
interface
uses
 DSharp.PresentationModel;
type
 [InheritedExport]
 IColorViewModel = interface
 ['{BCF3E6B6-2684-4D04-99D7-B2E05400A6C4}']
 end;
 [InheritedExport]
 IShellViewModel = interface
 ['{04C6473A-7E92-4ED1-B9A1-2B07D65277DC}']
 end;
...
end.
```

## Example: MindScape Color – part C; 4/...

4. Make your ShellView's
  - TPanel refer to the ColorModel
  - TShape's Brush.Color refer to Color

```
unit ShellViewForm;
interface
...
uses
...
 DSharp.PresentationModel, DSharp.Bindings.VCLControls;
type
 TShellView = class(TForm)
 ColorModel: TPanel;
 [Binding('Brush.Color', '{Binding Color}')]
 Shape1: TShape;
 end;
...
initialization
 TShellView.ClassName;
end.
```



### Example: MindScape Color – part C; 5/...

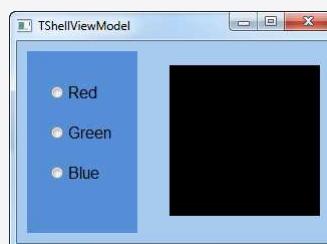
5. Make your ColorView have
- 3 RadioButtons for Red/Green/Blue
- ```

unit ColorViewFrame;
interface
uses
  Windows, Messages,
  SysUtils, Variants, Classes,
  Graphics, Controls, Forms, Dialogs, StdCtrls,
  DSharp.PresentationModel,
  DSharp.Bindings.VCLControls;
type
  TColorView = class(TFrame)
    Red: TRadioButton;
    Green: TRadioButton;
    Blue: TRadioButton;
  end;
implementation
{$R *.dfm}
initialization
  TColorView.ClassName;
end.
```



Example: MindScape Color – part C; 6/...

6. The goal is to
- Host the ColorView in the ShellView
 - Let the ShellViewModel accept events/messages about color changes in the ColorViewModel
 - Change Brush.Color of the TShape to the Color in the ShellViewModel
 - Have no code in the ColorView or ShellView



Example: MindScape Color – part C; 7/...

7. Create a ColorViewModel like this:

```

unit ColorViewModel;
interface
uses
  Graphics, Interfaces, DSharp.PresentationModel,
  DSharp.PresentationModel.EventAggregator;
type
  TColorViewModel = class(TScreen, IColorViewModel)
  private
    FEvents: IEventAggregator;
  public
    constructor Create(const Events: IEventAggregator);
    procedure Red;
    procedure Green;
    procedure Blue;
  end;
implementation
uses
  ColorEvent;
constructor TColorViewModel.Create(const Events: IEventAggregator);
begin
  inherited Create();
  FEvents := Events;
end;
procedure TColorViewModel.Red;
begin
  FEvents.Publish(TColorEvent.Create(clRed));
end;
...
initialization
  TColorViewModel.ClassName;
end.
```

Example: MindScape Color – part C; 8/...

8. Create a ShellViewModel interface like this:

```

unit ShellViewModel;
interface
uses
  Classes, SysUtils, Graphics,
  DSharp.PresentationModel,
  DSharp.PresentationModel.EventAggregator,
  Interfaces, ColorEvent;

type
  TShellViewModel = class(TScreen, IShellViewModel, IHandle<TColorEvent>)
  strict private
    FColor: TColor;
    FColorModel: IColorViewModel;
    procedure SetColor(const Value: TColor);
  public
    /// This constructor is called by the Dependency Injection container with
    /// parameters already created for you.
    constructor Create(const ColorModel: IColorViewModel;
      const Events: IEventAggregator);
    /// Implements IHandle<TColorEvent>.
    /// This method is called after a TColorEvent message is published from
    /// somewhere else in the application.
    procedure Handle(AMessage: TColorEvent);
    /// This property is for changing the color of the rectangle.
    property Color: TColor read FColor write SetColor;
    property ColorModel: IColorViewModel read FColorModel;
  end;
```

Example: MindScape Color – part C; 9/...

9. Create a ShellViewModel like this:

```

implementation

constructor TShellViewModel.Create(const ColorModel: IColorViewModel;
  const Events: IEventAggregator);
begin
  inherited Create();
  FColorModel := ColorModel;
  // Get the event aggregator through the constructor and
  // subscribe this ColorViewModel so it can listen for ColorEvent
  // messages.
  Events.Subscribe(Self);
end;

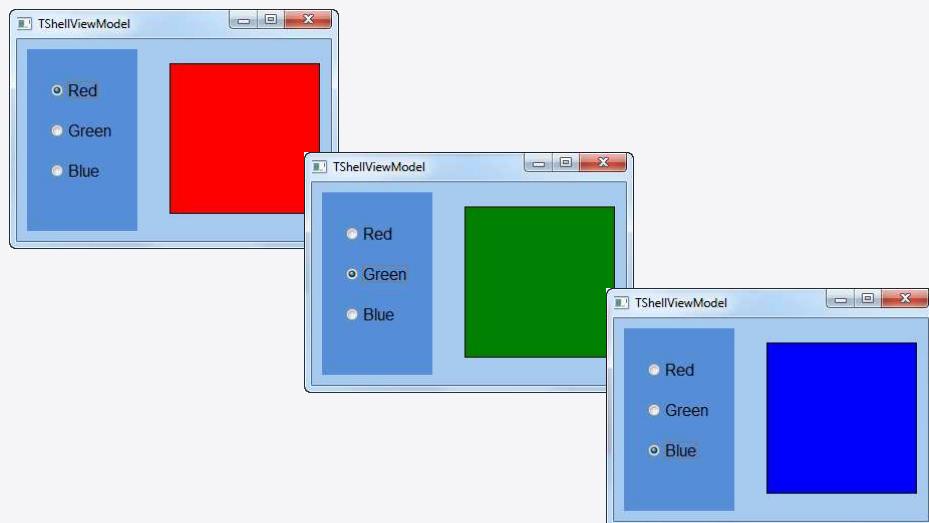
procedure TShellViewModel.Handle(AMessage: TColorEvent);
begin
  Color := AMessage.Color;
end;

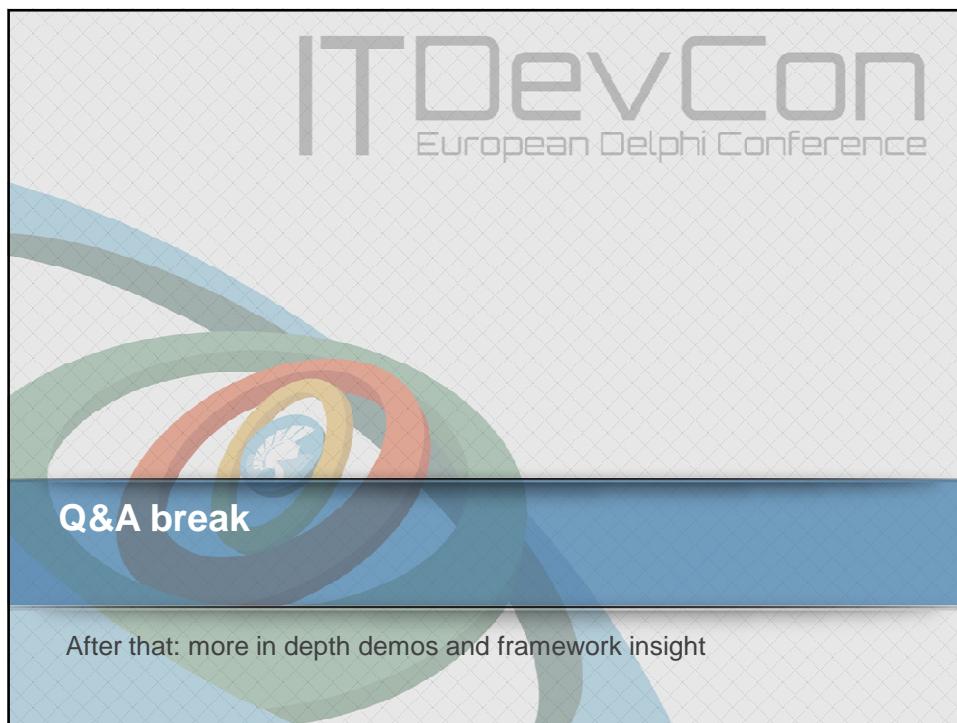
procedure TShellViewModelSetColor(const Value: TColor);
begin
  FColor := Value;
  NotifyOfPropertyChanged('Color');
end;

initialization
  TShellViewModel.ClassName;
end.
```

Example: MindScape Color – part C; 10/...

10. Run.





The slide has a blue header bar with the text 'DataContext / OnDataContextChanged' on the left and the 'IT DevCon' logo on the right. The main content area is white and contains a bulleted list with one item. The code for the 'OnDataContextChanged' event is shown in a monospaced font.

- DataContext / OnDataContextChanged

```
unit DSharp.Core.Extensions;
...
type
  TComponentHelper = class helper for TComponent
  ...
public
  procedure ClearValue(Prop: TDependencyProperty);

  function GetValue(Prop: TDependencyProperty): TValue;
  procedure SetValue(Prop: TDependencyProperty; const Value: TValue);

  property DataContext: TObject
    read GetDataContext write SetDataContext;

  property IsComponentInitialized: Boolean
    read GetIsComponentInitialized
    write SetIsComponentInitialized;

  property OnDataContextChanged: IEvent<TPropertyChangedEvent>
    read GetOnDataContextChanged;
end;
```

DependencyProperty

- DependencyProperty
 - Property that is not part of the object instance
 - Through a `helper` can be accessed like it is part of that instance
 - Has built-in binding support
- Delphi uses something similar for the
 - TFlowPanel
 - Adds a `ControlIndex` property to all controls on the panel
 - TGridPanel
 - Adds these properties to all controls on the panel
 - Column
 - Row
 - ColumnSpan
 - RowSpan

Binding through notifications

- NotifyOfPropertyChanged from ViewModel to View:


```
procedure TShellViewModel.SetColor(const Value: TColor);
begin
  FColor := Value;
  NotifyPropertyChanged('Color');
end;
```

 - Inside Caliburn, it gets translated to a DoPropertyChanged running on the UI thread by `TPropertyChangedBase.NotifyPropertyChanged`
- From View to ViewModel:
 - unit DSharp.Bindings.VCLControls has interceptor classes:


```
procedure TEdit.Change;
begin
  inherited;
  NotifyPropertyChanged.DoPropertyChanged('Text');
end;
```

Demos

IT DevCon
European Delphi Conference

- AppView VCL
- AppView DUnit
- Color
- HowToOpenDialog
 - IoC
<http://caliburnmicro.codeplex.com/wikipage?title=The%20Service%20Locator&referringTitle=Documentation>
 - WindowManager
<http://caliburnmicro.codeplex.com/wikipage?title=The%20WindowManager&referringTitle=Documentation>
- EventAggregatorSample
- SimpleDependencyProperty

IT DevCon
European Delphi Conference

14, 15 november 2013 - VERONA [Italy]

Current state of affairs



- workable alpha state
- beta release aimed in Spring 2014

bit Time software



Caliburn / Caliburn.Micro references

IT DevCon
European Delphi Conference

- Start at <http://caliburnmicro.codeplex.com/documentation>
- Mindscape intro:
 - <http://www.mindscapehq.com/blog/index.php/2012/01/12/caliburn-micro-part-1-getting-started/>
 - <http://www.mindscapehq.com/blog/index.php/2012/1/16/caliburn-micro-part-2-data-binding-and-events/>
 - <http://www.mindscapehq.com/blog/index.php/2012/01/24/caliburn-micro-part-3-more-about-events-and-parameters/>
 - <http://www.mindscapehq.com/blog/index.php/2012/02/01/caliburn-micro-part-4-the-event-aggregator/>
 - <http://www.mindscapehq.com/blog/index.php/2012/03/13/caliburn-micro-part-5-the-window-manager/>
 - <http://www.mindscapehq.com/blog/index.php/2013/09/11/caliburn-micro-part-6-introduction-to-screens-and-conductors/>
- Lots of references and articles:
 - <http://karlshifflett.wordpress.com/archive/mvvm/>
- Caliburn Micro and Windows RT
 - <http://www.terrymarshall.com.au/Blog/tabid/162/tagid/37/Caliburn-Micro.aspx>
- Caliburn Micro and Windows Phone 8
 - <http://wp.qmatteoq.com/first-steps-with-caliburn-micro-in-windows-phone-8-the-complete-series/>

Cross platform...

ITDevCon
European Delphi Conference

- GitHub Halp App:
Minimizing Platform-specific code with MvvM
 - Slides
 - <http://www.slideshare.net/Xamarin/git-hub-halp-app-minimizing-platformspecific-code-with-mvvm-by-justin-spahrsummers>
 - Video
 - <http://xamarin.com/evolve/2013#session-zm59b5yptf>

ITDevCon
European Delphi Conference



The end...
Actually: Questions & Answers

Blog: <http://wiert.me> Downloads: <https://bitbucket.org/jeroenp/conferences>