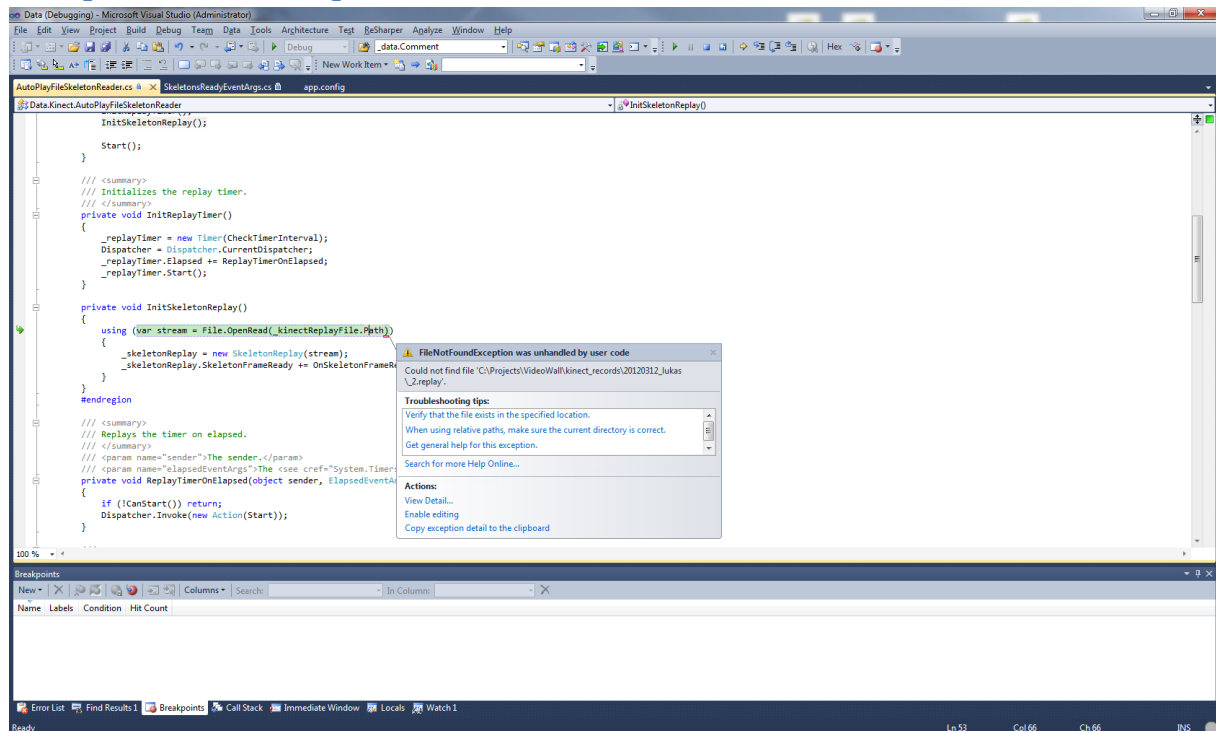


## Code Review vom 03.05.2012

### Exception bei Startup

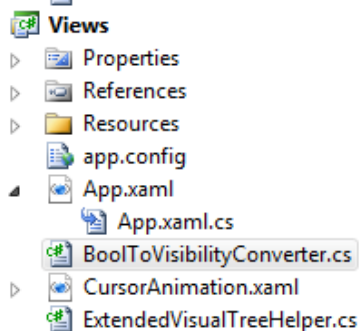


Could not find file

'C:\\Projects\\VideoWall\\kinect\_records\\20120312\_lukas\\\_2.replay'.': "C:\\Projects\\VideoWall\\kinect\_records\\20120312\_lukas\\\_2.replay"

- ➔ ZIP-Extract notwendig (mit Lukas besprochen?), ist dieses Vorgehen auch in der Doku beschrieben?
  - Problem: Nicht eingeweihte bzw. neu zum Projekt stossende Programmierer werden mit Garantie in das Problem laufen.
  - Daher: Bitte Deployment anders lösen (z.B. mit Resources), der Code darf (auch beim Entwicklungszeitpunkt) keine Pfade enthalten (DITO in LunchMenuReader).

### Organisation der Namespaces



Converter(s) bitte in eigenes Package / Ordner.

## Image

Tipp: `RenderOptions.BitmapScalingMode="HighQuality"`

Sieht besser aus und es Laggt weniger (!)

### Ergebnis

#### Testbed

- Kommunikation mit Client über einzelnen WCF Service
- Datenbankzugriff über Linq to SQL
- ASP Session als einzige Sicherheit

#### User Interface

- Selbsterklärend
- Trennung durch MVVM
- Auslagerung auf GPU

### Ergebnis

#### Testbed

- Kommunikation mit Client über einzelnen WCF Service
- Datenbankzugriff über Linq to SQL
- ASP Session als einzige Sicherheit

#### User Interface

- Selbsterklärend
- Trennung durch MVVM
- Auslagerung auf GPU

## Scope & Utils

```
static internal class ExtendedVisualTreeHelper
{
    public static IEnumerable<TSearch> FindInVisualTreeDown<TSearch>(DependencyObject obj) where TSearch : class
    {
        if (obj == null) yield break;
        var typedObj = obj as TSearch;
        if (typedObj != null) yield return typedObj;

        for (var i = 0; i < VisualTreeHelper.GetChildrenCount(obj); i++)
        {
            var children = FindInVisualTreeDown<TSearch>(VisualTreeHelper.GetChild(obj, i));
            foreach (var child in children.Where(child => child != null))
            {
                yield return child;
            }
        }
    }
}
```

static internal class / public static method -> Semantik bewusst eingesetzt?

Util's bitte in eigenes Package, evtl. sogar in Common GUI-Library?

Kommentare im Beispiel fehlen.

## Extension Methods

Bitte Information Expert einhalten: Das folgende Code-Stück wäre ein gutes Beispiel für eine Extension-Method auf `Button->Click()` :

```

/// <summary>
/// Raises the event of UI element.
/// </summary>
/// <param name="uiElement">The UI element.</param>
private void RaiseEventOfUIElement(Button uiElement)
{
    var peer = new ButtonAutomationPeer(uiElement);
    var invokeProv = peer.GetPattern(PatternInterface.Invoke) as IInvokeProvider;
    Debug.Assert(invokeProv != null, "invokeProv != null");
    invokeProv.Invoke();
}

```

## MVVM

UI Elemente im ViewModel ist unschön. Zerstört die Testbarkeit.

## Logger

Logger (log4net) fehlt (noch)? Console.WriteLine() im Code sehr unschön, sobald der Code auf Build Servern ausgeführt wird.

```

Started += (sender, args) => Console.WriteLine("UIElement " + args.UIElement + " started!");
Stopped += (sender, args) => Console.WriteLine("UIElement " + args.UIElement + " stopped!");
Clicked += (sender, args) => Console.WriteLine("UIElement " + args.UIElement + " clicked!");

```

## Memory-Problem

OutOfMemoryException ist beim Rumspielen aufgetreten. Vorsicht: Mengengerüst besagt, dass viel mehr Daten (Posters) geladen werden müssen. Ist dies in der Risiko-Analyse beschrieben?

Die Posters werden auf einem Intel Core i5 660 / 4GB Ram nur mit „LAG“ geladen. Dies könnte ebenfalls ein erweitertes Problem darstellen, falls die Daten übers Netzwerk geladen werden (dann z.B. lokal kopieren)?

An diversen Stellen werden die Bilder mit einem File-Stream geöffnet:

```
public static class ImageExtension
{
    /// <summary>
    /// Initializes the BitmapImage with the string.
    /// </summary>
    /// <param name="img">The image.</param>
    /// <param name="fileName">Name of the file.</param>
    /// <returns></returns>
    public static BitmapImage InitializeWith(this BitmapImage img, string fileName)
    {
        img.BeginInit();
        img.StreamSource = File.OpenRead(fileName);
        img.CacheOption = BitmapCacheOption.None;
        img.EndInit();
        img.Freeze();
        return img;
    }
}
```

Wer schliesst diesen Stream? Verursacht dies ein Memory-Leak? Before/After Pattern nicht angewendet/eingehalten?

Unschön: Konzept des Memory-Managments (Caller / Callee konsumiert Stream) nicht in der Code-Dokumentation der Methode festgehalten. Dies stellt ein potentielles Problem für Memory-Leaks dar.

Besser direkt als Initialisierungs-Methode: LunchMenu::CreateFrom(string fileName) ?

```
name = "LunchMenu::CreateFrom(string fileName)";
Image = new BitmapImage();
Image.InitializeWith(fileName);
```

## Domain Objekte / Business Layer

Wo befindet sich das ServiceModel (bzw. die Domain Objekte)?

Nun greift das ViewModel direkt auf die DatenServices (DL) zu. Betreffend Nomenklatur ist dies irreführend oder gewollt?

## Dispose() Pattern

Dieses wurde an diversen Stellen nicht vollständig implementiert, z.B.:

```
class KinectSkeletonReader : ISkeletonReader
{
    private readonly KinectSensor _kinectSensor;
    private Skeleton[] _skeletons;
    public event EventHandler<SkeletonsReadyEventArgs> SkeletonsReady;
```

ISkeletonReader ist zwar IDisposable, allerdings fehlt die komplette Implementation des Dispose-Patterns (siehe [http://msdn.microsoft.com/en-us/library/b1yfk5e\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/b1yfk5e(v=vs.71).aspx) )

## WPF

```
// VisualTreeHelper.HitTest(Application.Current.MainWindow, _cursorViewModel.Position);
foreach (var element in HitElements)
{
    var relativePosition = _window.TranslatePoint(_cursorViewModel.Position, element);
    var hit = element.InputHitTest(relativePosition) != null;

    // if the current element is the same as the element but that one was not hit, stop the timer
    if (_currentElement == element && !hit) { StopTimerAndResetCurrentElement(); continue; }

    // if the button was not hit, move to the next element
    if (!hit) continue;

    // if the current element is the hit element, than the timer has been started already
    if (_currentElement == element) continue;

    // if a new element is hit, the old timer must be stopped
    if (_currentElement != null) StopTimerAndResetCurrentElement();

    StartTimerAndSetCurrentElementTo(element);
}
```

Weshalb ein Loop und nicht die HitTest Methode (1. Linie)?

ViewModel Verschachtelung, MenuViewModel zu mächtig bzw. weiss mehr als MainWindowViewModel? Warum so gelöst?

```
<Tool:ReportControl Grid.Row="0" x:Name="Report1">
    <UserControls:ScrumUserControl.BuildUpViewModelSource>
        <Tools:ViewModelReportControl Name="Daily_Report_1"></Tools:ViewModelReportControl>
    </UserControls:ScrumUserControl.BuildUpViewModelSource>
</Tool:ReportControl>
```

PropertyChanged vs Changes mit Events.

Warum Thread based Timer und nicht DispatcherTimer?

## Debugging

Wozu dieser Assert hier? Muss die Funktionalität des .NET Frameworks nochmals überprüft werden?

```
Debug.Assert(invokeProv != null, "invokeProv != null");
invokeProv.Invoke();
```

## **Dokumentation**

Entwurf.docx -> Converter -> Converter (beispielsweise zur Konvertierung von bool zu visibility)  
-> Ist im View?

Technische Dokumentation nicht bis zum Ende aufschieben. Auch Erkenntnisse von den Prototypen?

## **Generelle Anmerkungen zum Programm-Code**

Regionen abgeschafft / nicht geplant?

Reihenfolge und Strukturierung von Properties / Methoden / Constructoren ist inkonsistent.

Code-Guidelines vorhanden & eingehalten?

Keine Preconditions eingesetzt (haben wir doch besprochen)?