Можно ли жить с UWP?

Никита Каменский, Tinkoff

Universal Extensions

- Desktop
- Mobile
- IoT
- Xbox

…and one app to rule them all

UWP Versions

- Build 10240 — initial release
- Build 10586 — Windows Hello, Composition
- Build 14393 (Anniversary) — Windows Ink, Cortana APIs
- Build 15063 (Creators) — new Composition APIs, Payments

Minimum Version vs Target Version

Adaptive Code

```
ApiInformation.IsTypePresent(“Windows.Phone.UI.Input.HardwareButtons")

ApiInformation.IsMethodPresent(
    “Windows.ApplicationModel.Calls.PhoneCallManager",
    “ShowPhoneCallUI",
    2)

ApiInformation.IsEventPresent(
    “Windows.Phone.UI.Input.HardwareButtons",
    "BackPressed")
```

.NET Native

1. MSIL
2. Generating interop marshaling and serialization code
3. Merging
4. Reducing
5. MSIL transformations
6. MSIL -> Machine Dependent Intermediate Language
7. MDIL -> native code

.NET Native Random Facts

- no AnyCPU build configuration
- F# is not supported
- but System.Numerics.Vectors namespace is!

.NET Native vs NGEN

- NGEN falls back to JITing code if no native image is available, .NET Native produces only native images
- Change of dependency requires dependants to be reNGENed, .NET Native applications are served independently

Runtime Directives

```
<Directives xmlns="http://schemas.microsoft.com/netfx/2013/01/metadata">
  <Application>
    <Assembly Name="*Application*" Dynamic="Required All" />

    <Assembly Name="System.Collections.NonGeneric">
      <Type Name="System.Collections.ArrayList" Dynamic="Required All"/>
    </Assembly>

    <TypeInstantiation Name="Microsoft.EntityFrameworkCore.ChangeTracking.Internal.Snapshot"
                       Arguments="System.String,System.String,System.DateTimeOffset,System.Double"
                       Activate="Required Public" />

    <Type Name="System.Data.Common.DbDataReader">
      <MethodInstantiation Name="GetFieldValue"
                           Arguments="System.DateTimeOffset"
                           Dynamic="Required"/>
    </Type>

  </Application>
</Directives>
```

{x:Bind}

- Markup extension
- Compile-time validation
- ~~UpdateSourceTrigger~~
- ~~Binding to Source~~

{x:Bind}

```
<TextBlock Text="{x:Bind Numbers[1]}” />

<TextBlock Text="{x:Bind OnlineBanks[‘Tinkoff']}" />

<TextBlock Visibility="{x:Bind ConvertBoolToVisibility(IsVisible),
                              BindBack=ConvertVisibilityToBool}" />
```

x:Phase

```
<StackPanel Orientation="Horizontal">
    <Image Source="{x:Bind CardImage}" x:Phase="2"/>
    <TextBlock Text="{x:Bind CardName}" FontSize="12"/>
    <TextBlock Text="{x:Bind Balance}" x:Phase="1"/>
</StackPanel>
```

VisualState.Setters

```
<VisualState x:Name="PointerOver">
    <Storyboard>
        <ObjectAnimationUsingKeyFrames Storyboard.TargetName="GlyphElement"
                                       Storyboard.TargetProperty="Foreground">
            <DiscreteObjectKeyFrame KeyTime=“0" Value=“#00000000" />
        </ObjectAnimationUsingKeyFrames>
    </Storyboard>
</VisualState>
```

VisualState.Setters

```
<VisualState x:Name="PointerOver">
    <VisualState.Setters>
        <Setter Target="GlyphElement.Foreground" Value="#FFFFFFFF" />
    </VisualState.Setters>
</VisualState>
```

AdaptiveTrigger

```
<VisualState x:Name="Mobile">
    <VisualState.StateTriggers>
        <AdaptiveTrigger MinWindowWidth="0" />
    </VisualState.StateTriggers>
    <VisualState.Setters>
        <Setter Target="PageHeader.Padding" Value="12" />
    </VisualState.Setters>
</VisualState>

<VisualState x:Name="Wide">
    <VisualState.StateTriggers>
        <AdaptiveTrigger MinWindowWidth="500" />
    </VisualState.StateTriggers>
    <VisualState.Setters>
        <Setter Target="PageHeader.Padding" Value="24, 12" />
    </VisualState.Setters>
</VisualState>
```

x:DeferLoadingStrategy

```
<Grid>
    <Grid x:Name="MobilePanel"
          x:DeferLoadStrategy="Lazy"
          Visibility="Collapsed">
        <!-- Mobile specific UI -->
    </Grid>

    <Grid x:Name="DesktopPanel"
          x:DeferLoadStrategy="Lazy"
          Visibility="Collapsed">
        <!-- Desktop specific UI -->
    </Grid>
</Grid>
```

x:DeferLoadingStrategy

```
<VisualState x:Name=“Wide">
    <VisualState.Setters>
        <Setter Target="DesktopPanel.Visibility" Value="Visible" />
    </VisualState.Setters>
</VisualState>

<VisualState x:Name="Mobile">
    <VisualState.Setters>
        <Setter Target=“MobilePanel.Visibility" Value="Visible" />
    </VisualState.Setters>
</VisualState>
```

Demo Time

Statistics Time

- Those who develop for Android mostly have Android smartphones (77%), while those who develop for iOS have Apple iOS smartphones (64%).
- Only 17% of those who develop for Windows have Windows-based smartphones.

Ein Code, Ein Programmierer, Ein Benutzer