

Univerzális Windows Alkalmazások

Fekete Krisztián

december 1. (hétfő), 12:00-15:45



Wi-Free

- SSID: visitor
- WPA jelszó: swvk5iru45jbmvsv

Tartalom

- Univerzális Windows alkalmazások
 - > Kronológia
 - > Architektúra
 - > Platformszolgáltatások
 - > Alkalmazás fejlesztési modell
 - > Közös alkalmazásfejlesztés
- .NET 2015
 - > Újdonságok
 - > Visual Studio 2015, Blend 2015
- Windows 10
 - > Bemutató



Univerzális Windows alkalmazások

Kronológia, Architektúra, Platformszolgáltatások,
Alkalmazás fejlesztési modell, Közös
alkalmazásfejlesztés

Kronológia 1

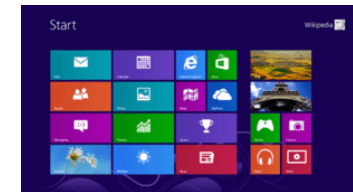
- Az elmúlt **5 év (!)** története



02 / 2010 (6.5.3)



11 / 2010



08 / 2012



10 / 2012



10 / 2013



10 / 2013



2015

Kronológia 2

Windows Phone 7.x



Windows Phone 8.0



Windows Phone 8.1

Konvergencia kezdete: IE9

- WP 7.5 IE9-el érkezik
- IE kódbázis Windows-al közös
- JavaScript engine ugyanaz mint a Windows-ban

Alacsony szintű konvergencia Core OS szinten

- Közös NT kernel, file rendszer és hálózatkezelés, secure boot, BitLocker

Fejlesztői platform

- API egy része már közös
- Natív kód (C++) és DirectX
- IE10

Fejlesztési platform konvergálása

- Kód és fejlesztési módszer újra felhasználása

Store-ok egységesítése

- Közös regisztráció és jogok

Magas szintű konvergencia Core OS szinten

- Proximity & Location
- Security & identity
- Background Tasks

Windows 7



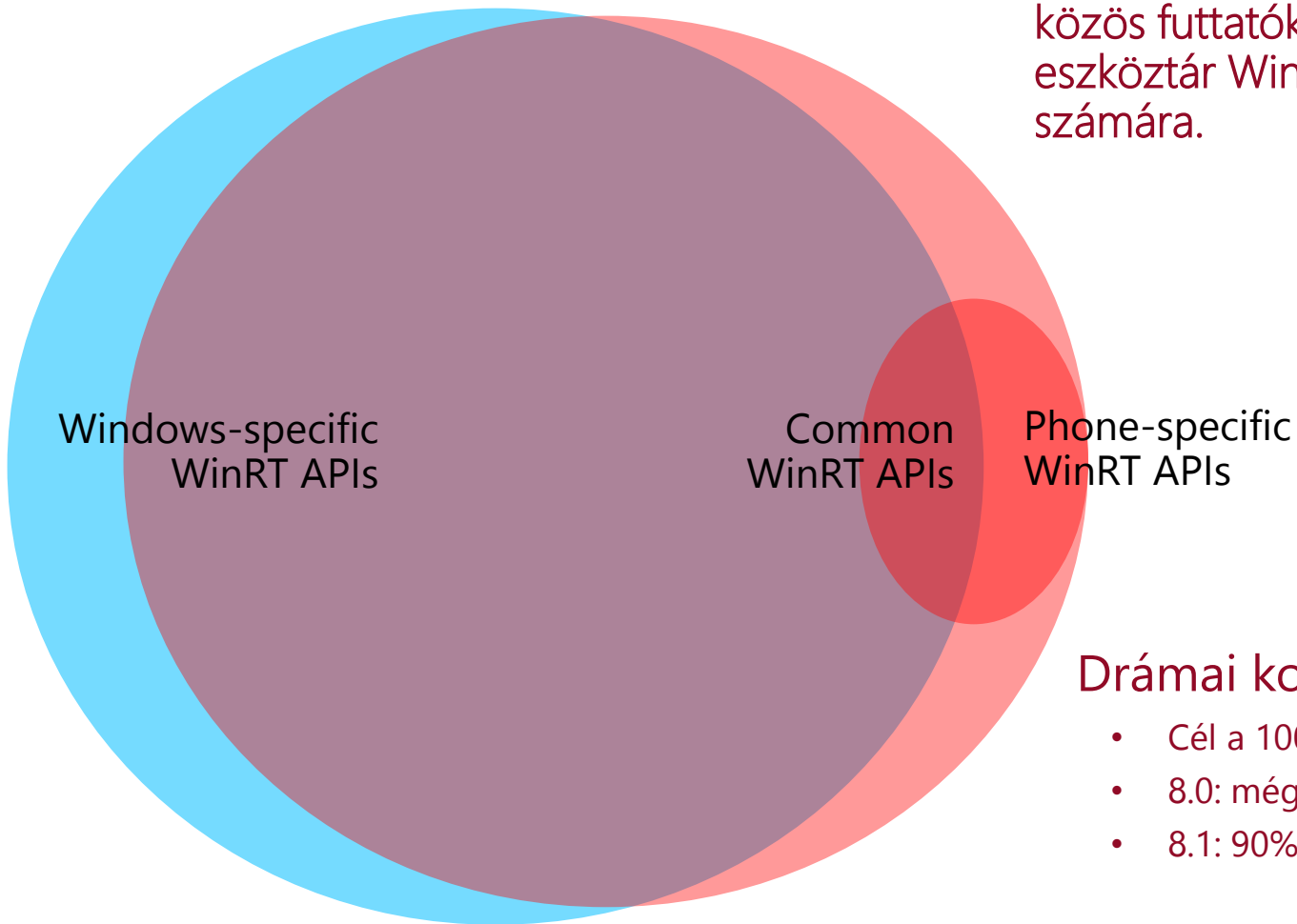
Windows 8.0



Windows 8.1

Architektúra - Windows 8.1 és Phone API konvergencia

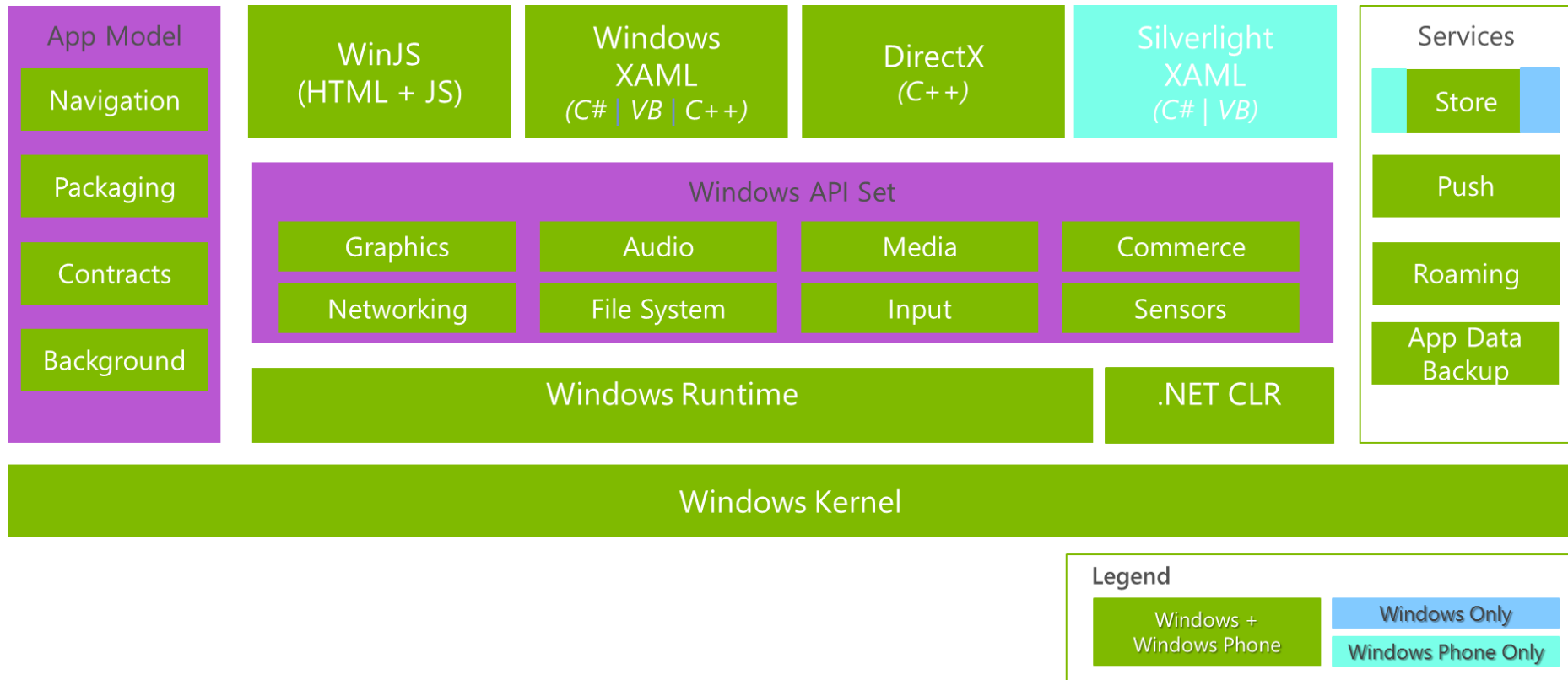
A Windows Runtime (WinRT) egy közös futtatókörnyezet és API eszköztár Windows Store alkalmazások számára.



Drámai konvergencia a 8.1-ben

- Cél a 100%-os fejlesztési konvergencia
- 8.0: még csak 30%-os konvergencia
- 8.1: 90%+-os konvergencia

Architektúra - Windows 8.1 és Phone API konvergencia



Ismétlés, alapok villámgyorsan!

- C# vagy VisualBasic (VB)
+ XAML- eXtensible Application Markup Language
- XML alapú deklaratív felület leíró nyelv
- Fejlesztők és a dizájnerek közös nyelve
- C++ & HTML + JavaScript (WinJS) is támogatott

```
<Button Name="mybtn" Type="Tap" Content="Click" Click="btn_click;" />
```

```
function btn_click(Tap( object sender, GestureEventArgs e)
{
    var btn = document.getElementById(mybtn);
    btn.setAttribute("value", "Clicked!");
    mybtn.Content = "Tapped!";
    alert("Hello!");
    MessageBox.Show( "Hello!" );
}
```

Fejlesztőeszköz

- Visual Studio 2013 / 2015
- Visual Studio Community
 - > full-featured IDE for free
- Visual Studio Express with Update 4 for Windows



<http://www.visualstudio.com/downloads/download-visual-studio-vs>

Alkalmazás fejlesztési modell

- Mi az alkalmazás fejlesztési modell?
 - > Deployment
 - > Execution
 - > Storage
 - > Navigation
 - > OS Integration
 - > Notifications

Mik az Univerzális alkalmazások?

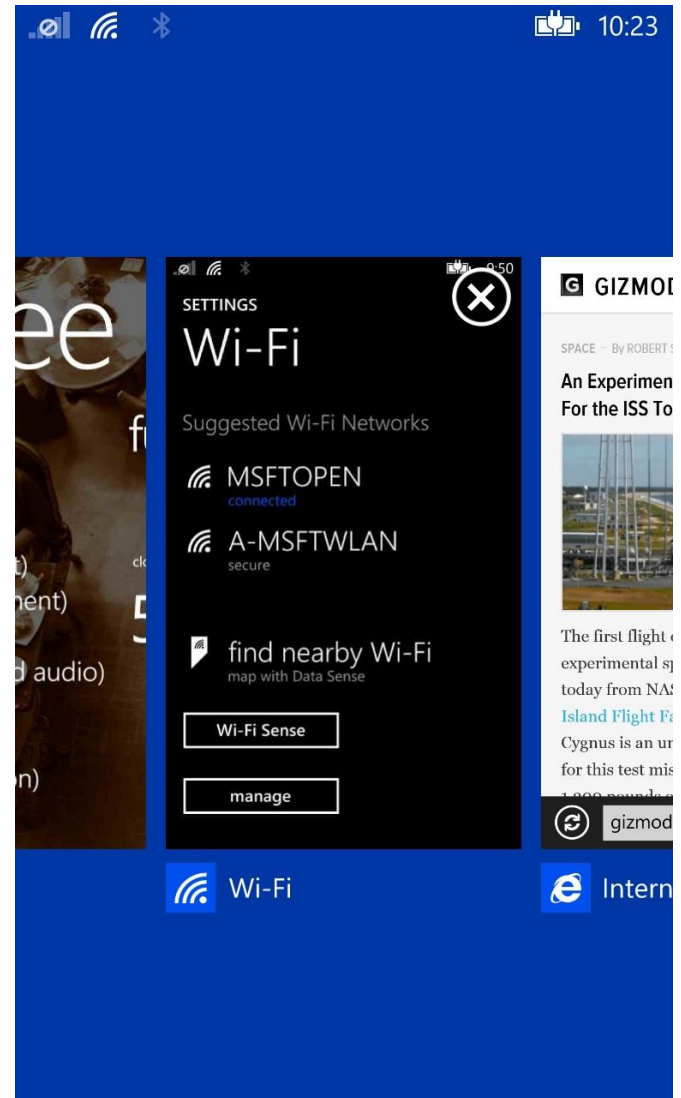
- „Unified”-nak is nevezik (talán találóbb)
- Megteremti annak a lehetőségét, hogy Windows és Windows Phone alkalmazásokat a legalacsonyabb szinten közös kóddal fejlesszünk
- **Ez egy lehetőség, tehát nem kötelező!**
- Támogatott nyelvek:
- Windows Runtime XAML +
 - > C#, VB, C++
- HTML + JavaScript / WinJS

Platformszolgáltatások

- Resume Behavior

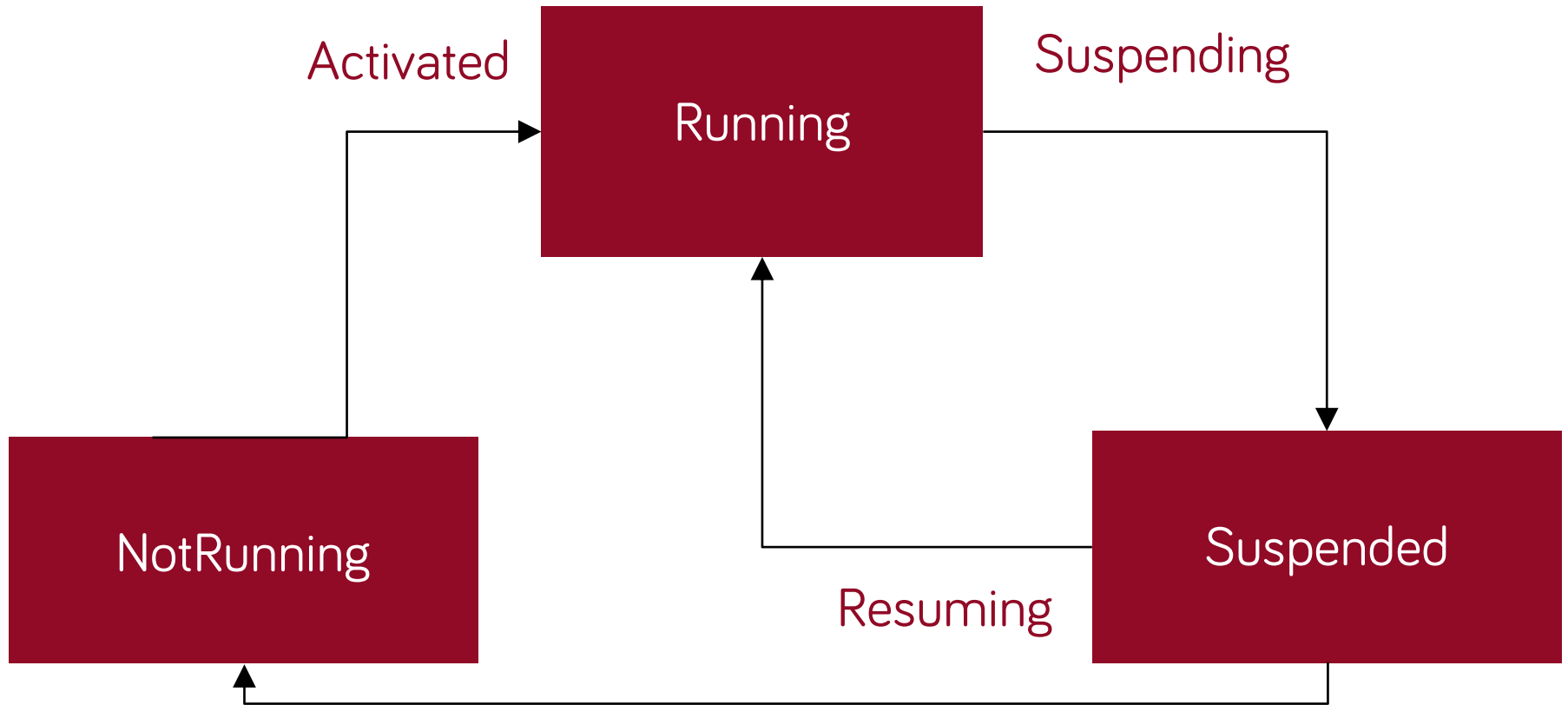
A Windows Phone 7.x és 8.0-ban a BACK gombbal bezártuk az alkalmazást

A Windows Phone 8.1-től kezdve a BACK soha nem zárja be az alkalmazást, az csak az App Switcher-rel lehetséges



Platformszolgáltatások

- Életciklus



Platformszolgáltatások

- Memória allokáció (WP8)
 - > Jól futott: 512 MB-os eszközökön
 - > Nagyon jól futott: > 512 MB-os eszközökön
- Cél: folyékony User Experience



Platformszolgáltatások

- Memória allokáció (WP8)
- A memória limitek előre definiáltak voltak:
 - > 150 Mb: 512Mb-os eszközök
 - > 300 Mb: 1Gb-os eszközök
 - > 450 Mb: 2Gb-os eszközök
- Érdekesség, az alkalmazások
 - > $94\% \leq 150\text{MB}$, $86\% \leq 100\text{MB}$,
 $80\% \leq 80\text{MB}$, $70\% \leq 60\text{MB}$ memóriát fogyaszt
- Eredmény:
 - > Sok alkalmazás low-end eszközökön is 'tombstoned' állapotba került, pedig nem kellett volna.
 - > User experience nem megfelelő, hiszen folyton újraindultak a programok

Platformszoftvertársítások

- Memória allokáció (WP8.1)
- Dinamikus memória allokálás

A rendszer monitorozza hogy az egyes alkalmazások mennyi memóriát használnak

Ezen adatok alapján állítja be a memória limiteket

Azért ez már egy jó világ 😊

Eredmény

Több alkalmazás marad a memóriában

512 MB – sok alkalmazás

>1GB – szinte az összes

Platformszoolgáltatások

- Háttérfolyamatok /Background Tasks/
 - > Triggerek és különböző feltételek teljesülése esetén a háttérben lejátszódó folyamatok
 - Pl.: időzóna változás, internet kapcsolat elérése stb.
 - > További triggerek:
 - „Push to wake a task” – Push Notification hatására indul a TASK (pl.: Csempe frissítése, Action Center értesítés)
 - Geofencing
 - Bluetooth és szenzorok kezelése

Platformszolgáltatások

- Notifikációk
- Windows Notification Service (WNS)
 - > Windows és Windows Phone közös értesítési rendszere
 - > A korábbi Windows Phone-os értesítések is a WNS-en keresztül dolgozódnak fel (API módosítás nem szükséges)

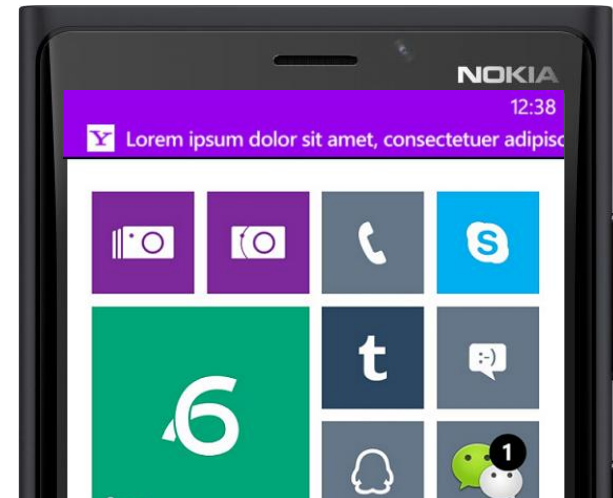
Eredmény: energiahatékonyabb és megbízhatóbb értesítési rendszer

Platformszoolgáltatások

- Notifikációk (Példa)

Közös adatformátum

Eszköz specifikus megjelenítés



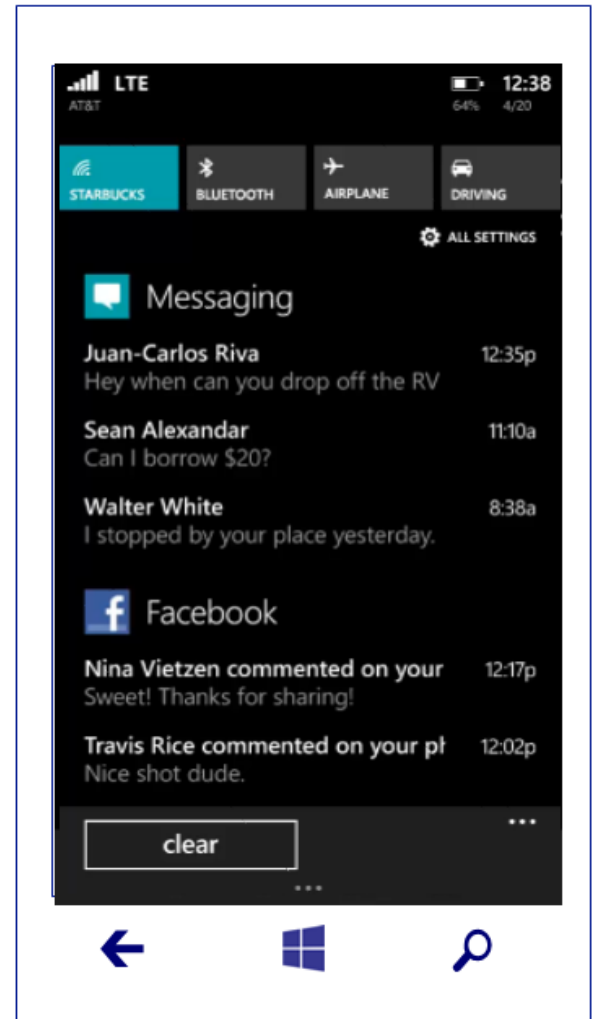
Platformszolgáltatások

- Action Center

A legfrissebb értesítések minden alkalmazástól egy helyen (mint Androidon és iOS-en)

Push Notification-on keresztül értesítés beszűrhető

Alkalmazások lokálisan is menedzselhetik az értesítéseket (törlés, hozzáadás)



Platformszolgáltatások

- Adattárolás és telepítés

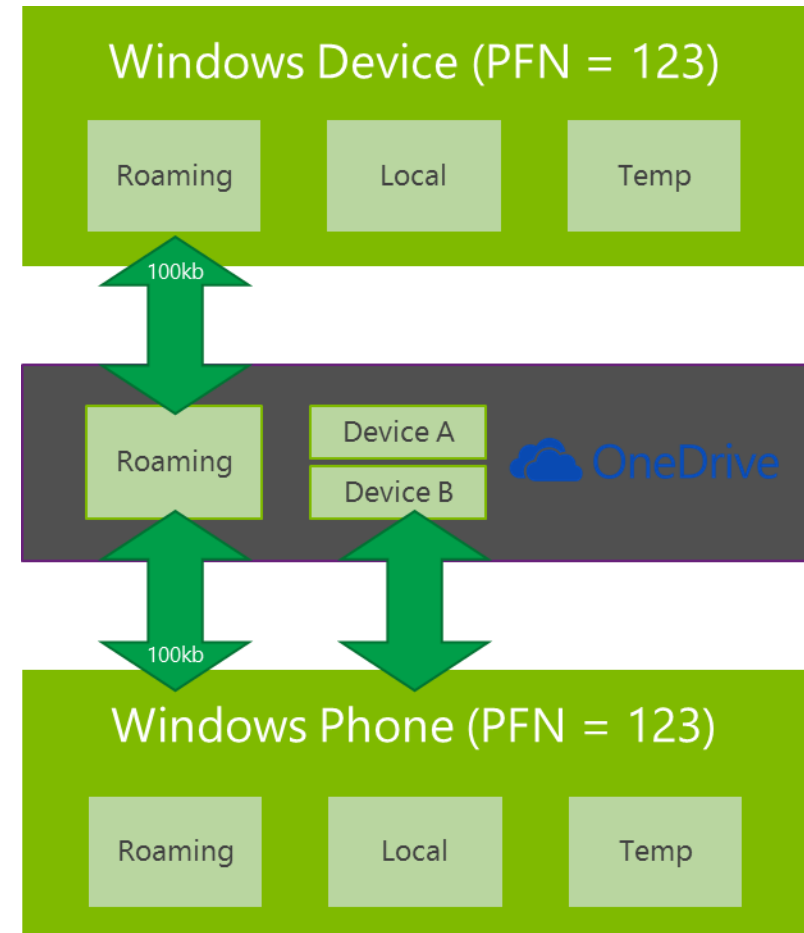
Azonos formátum:
Windows & Phone (APPX)

Bye-bye XAP



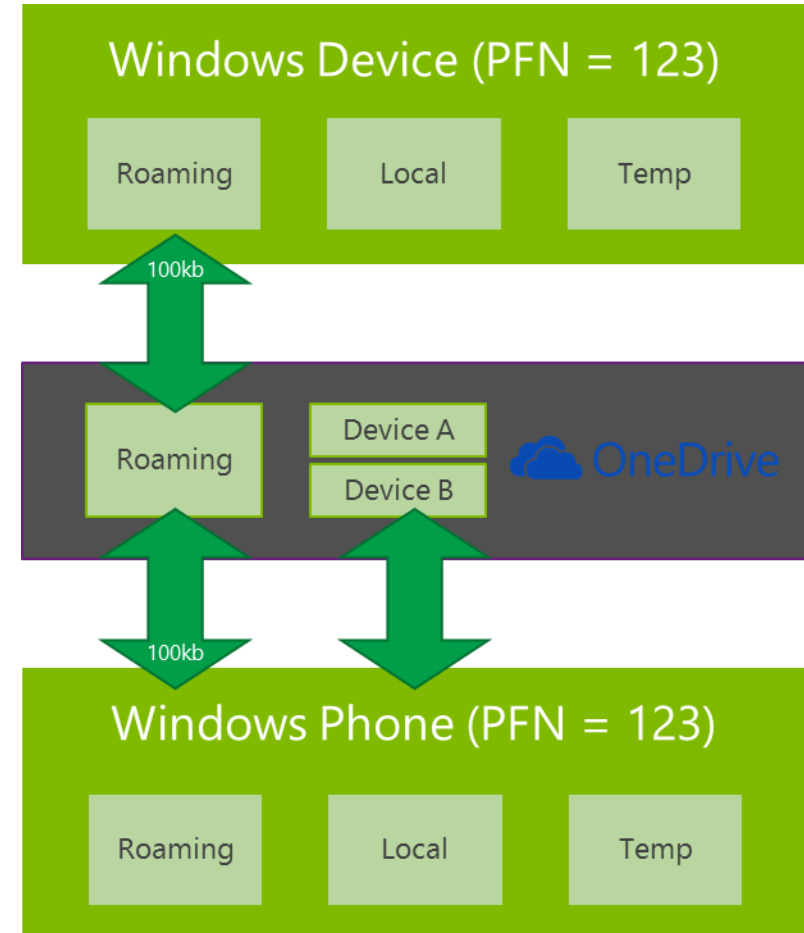
Platformszolgáltatások

- Eszközök közötti adat szinkronizálás
- 100 KB-nyi adat szinkronizálható
 - > Alkalmazás beállítások
 - > Félig kész munka stb.
- Az új adatokról a WNS értesíti az **aktív** alkalmazásokat



Platformszolgáltatások

- A lokális adatok (eszköz specifikus) is menthetők a OneDrive-ra de ott a felhasználói kvótából veszünk el tárterületet (100 KB-nál nem)



Mik az Univerzális alkalmazások?

- „Unified”-nak is nevezik (talán találóbb)
- Megteremti annak a lehetőségét, hogy Windows és Windows Phone alkalmazásokat a legalacsonyabb szinten közös kóddal fejlesszünk
- Ez egy lehetőség, tehát nem kötelező!
- Fejlesztési / architektúrális döntések!
- Universal != 1 app
 - > 2 alkalmazást írunk továbbra is!
- Támogatott nyelvek:
- Windows Runtime XAML +
 - > C#, VB, C++
- HTML + JavaScript / WinJS
- Cél: - One application, runs everywhere

Közös kódépítési stratégiák / Sharing strategies

- Mit oszthatok meg?

Windows Phone 8.1 App

Phone UI

Windows 8.1 App

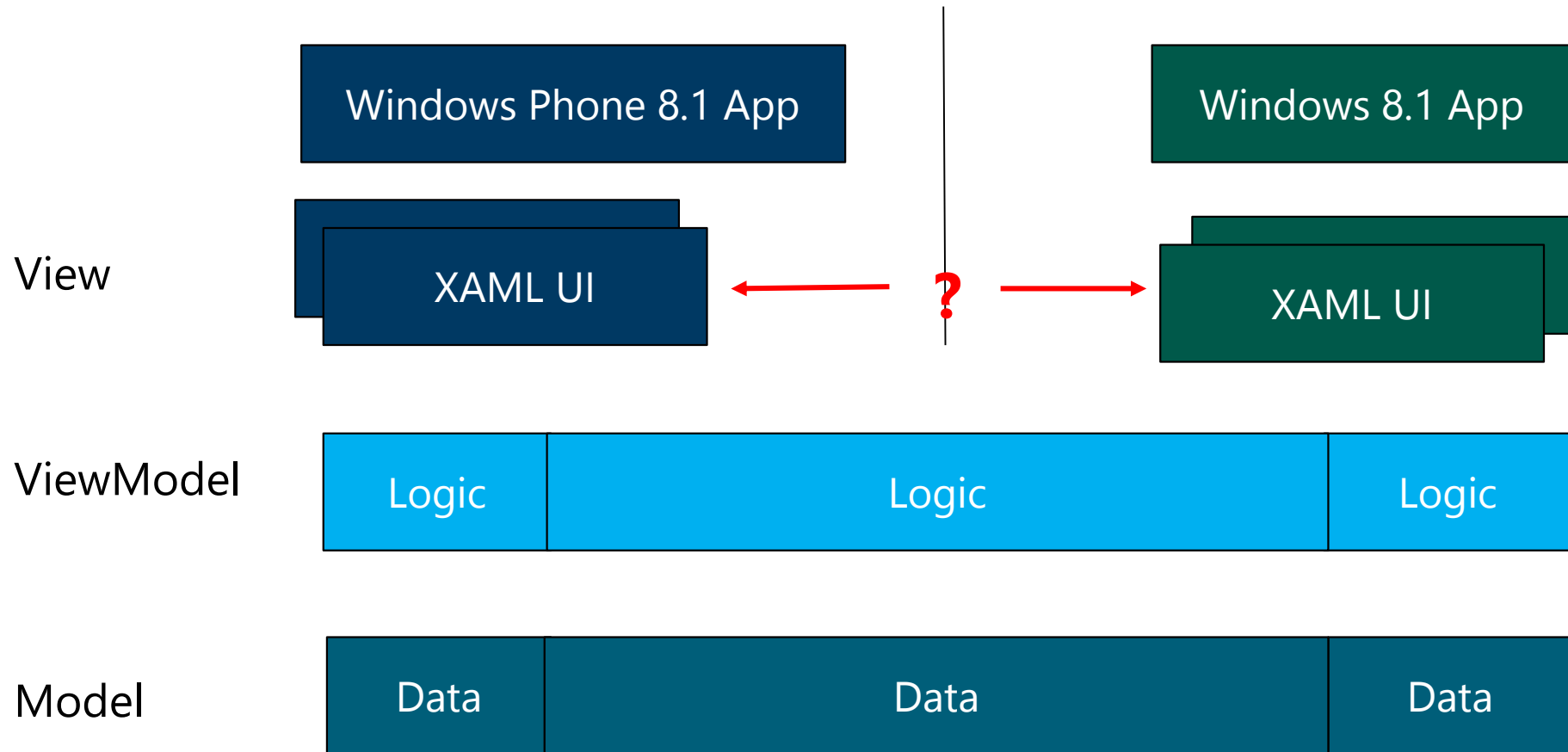
Windows UI

Shared Code, Images, Files

WinRT

Közös kódépítési stratégiák / Sharing strategies

- Mit oszthatok meg?



Kód megosztási lehetőségek / Code Sharing Options

- Shared Project Files
 - > Legtöbb fájl típus
- Portable Class Library
 - > Osztálykönyvtár & Windows Runtime komponens
- “Add As Link”

Shared Project

Forrás megosztása
Windows és WP projektek
között

Nincs bináris kimenet

Minden típus támogatott

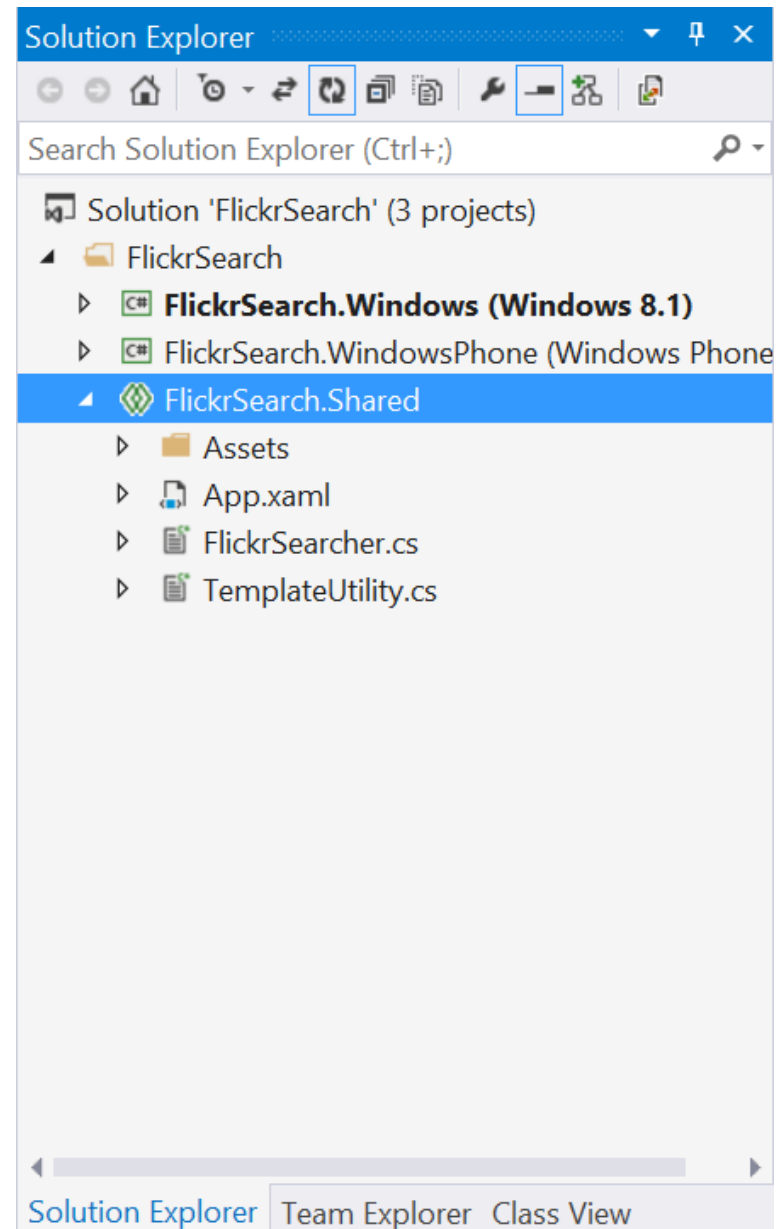
Kód fájlok

XAML

Képek

XML/JSON

RESX



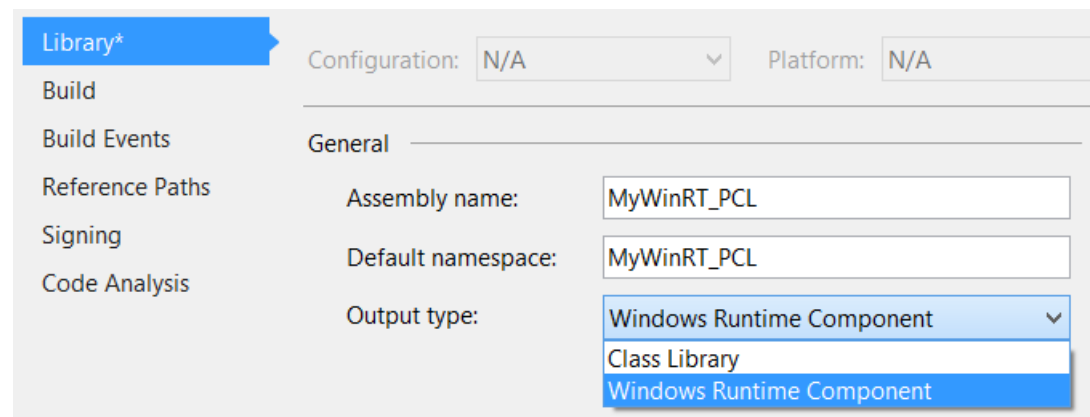
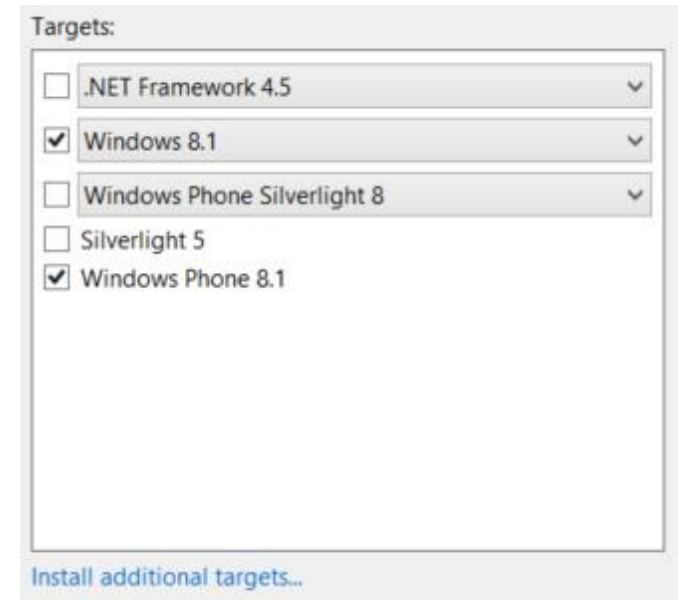
Portable Class Libraries

Windows Universal
alkalmazásoknak

WinRT API támogatása

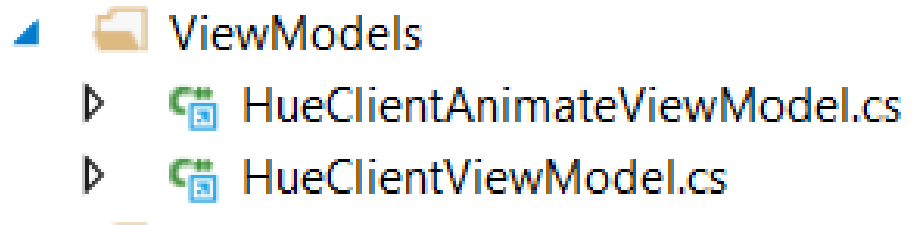
Output: Windows Runtime
komponens

C++, Javascript
alkalmazások



„Add As Link”

- A Shared projekthez hasonló megoldás
 - > Fordítási direktívák
 - > Előbbi kényelmesebb



Közös WinRT API használata

Windows 8.1

Windows Phone 8.1

Common WinRT APIs

Windows Only WinRT

- e.g. search contract
- e.g. multiple windows
- e.g. resizable windows
- e.g. printing support

Phone Only WinRT

- e.g. action center
- e.g. status bar
- e.g. back key handling

Közös API-k esetén is lehetnek eltérések (lásd később)

Egységes WinRT API Windows+Phone

- Példák:

files & settings: [local](#), [temp](#), [roaming](#), [pickers...](#)

store: [app purchases](#), [receipts...](#)

lifecycle: [launch](#), [suspend](#), [resume](#), [background tasks](#)

notifications: [tiles](#), [toasts](#), [badges](#), [push](#)

sensors: [gps](#), [geofencing](#), [gyro](#), [compass...](#)

network: [http](#), [websockets](#), [sockets...](#)

localisation: [resource resolution from XAML/code...](#)

Kivétel (példa 1)

- FileOpenPicker (,AndContinue' API-k)

Windows

```
//Create the picker object
FileOpenPicker openPicker = new FileOpenPicker();
openPicker.ViewMode = PickerViewMode.Thumbnail;
openPicker.SuggestedStartLocation =
    PickerLocationId.PicturesLibrary;

// Users expect to have a filtered view of their folders
openPicker.FileTypeFilter.Add(".jpg");
openPicker.FileTypeFilter.Add(".png");

// Open the picker for the user to pick a file
StorageFile file =
    await openPicker.PickSingleFileAsync();

if (file != null)
{
    // Do something with the file...
}
```

Windows Phone

```
//Create the picker object
FileOpenPicker openPicker = new FileOpenPicker();
// On Windows Phone, setting Filtering to image types
// causes Picker to show Camera Roll
openPicker.FileTypeFilter.Add(".jpg");
openPicker.FileTypeFilter.Add(".png");

// Open the picker for the user to pick a file
openPicker.PickSingleFileAndContinue();
```

App suspended, may be terminated



App activated when file picked

Kivétel (példa 2)

- Pozíció meghatározás, térkép

Geolocation 100%

Geofencing 100%

Geodata No: Location (Windows) vs. Geopoint (WP)

Map control No: Bing Maps (Windows) vs. WinRT Map control (WP)

Lehetséges ,Universal' app szinten kezelni, de a megjelenítésre, adat konverzióra megoldás kell (lásd közös fejlesztésű app!)

Platformok összeeresztése

- Technikák az egy kód / több platform problémára
 - > #if #else fordítási direktívák
 - > Öröklés
 - Absztrakt osztály / Interface
 - Dependency Injection
 - > Partial osztályok

#if #else fordítási direktívák

- Windows és Windows Phone megkülönböztetése

Windows = WINDOWS_APP

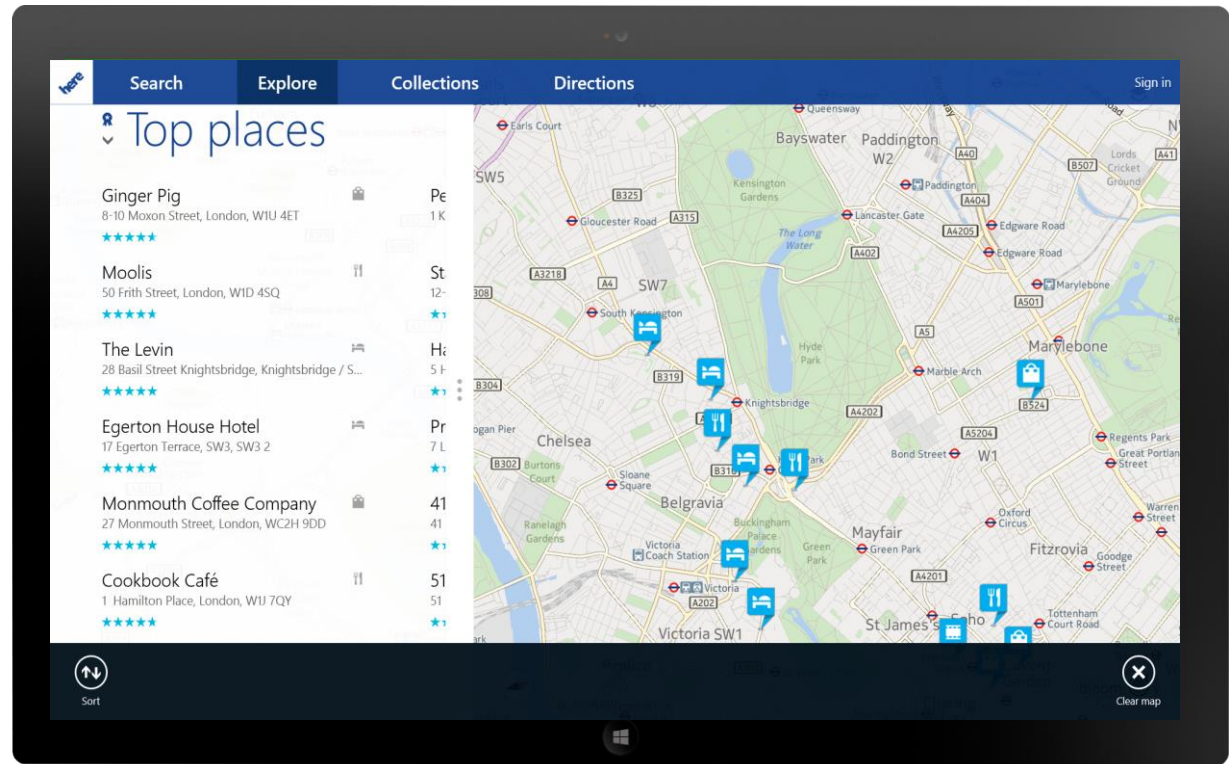
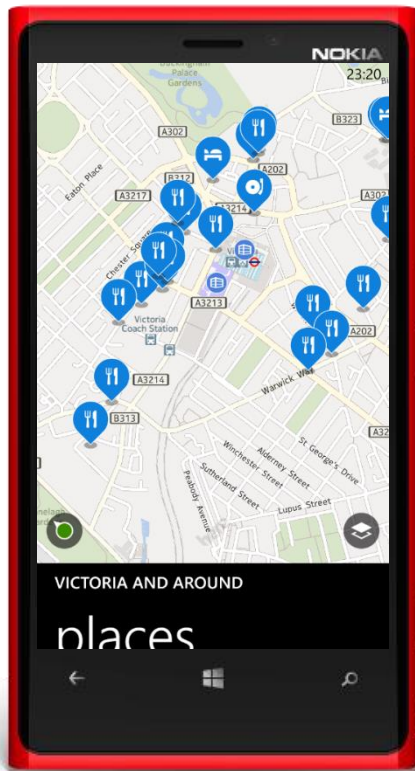
Windows Phone = WINDOWS_PHONE_APP

- Példa: Back gomb kezelése (csak Windows Phone)

```
#if WINDOWS_PHONE_APP
    Windows.Phone.UI.Input.HardwareButtons.BackPressed +=
        this.HardwareButtons_BackPressed;
#endif
```

XAML megosztása

- Lehetőség! Felhasználhatóság szempontból is mérlegelni kell!



XAML vezérlők Windows+Phone

common,
same rendering

Button



Slider



ToggleSwitch



ProgressBar



etc (many more)

common,
different content

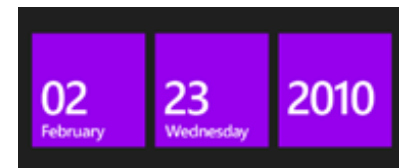
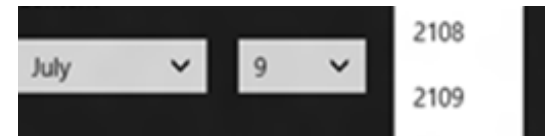
Hub



ListView
GridView
etc.

common,
different rendering

DatePicker

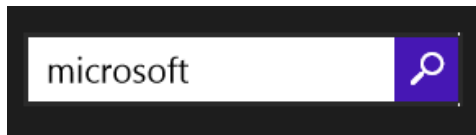


TimePicker
CommandBar
AppBar
etc.

XAML vezérlők Windows+Phone

unique

SearchBox



Pivot

ContentDialog

AutoSuggestBox

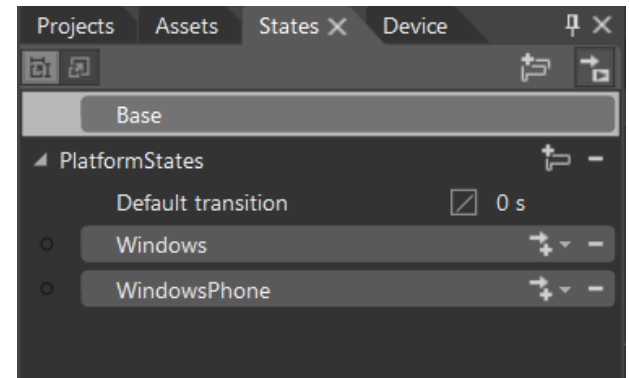
etc.

Közös XAML nézetek

- User Control
- Komplet XAML oldalak
- Platform specifikus nézetek.

1. Visual State Manager

```
#if WINDOWS_APP
    var result = VisualStateManager.GoToState(this, "Windows", false);
#elif WINDOWS_PHONE_APP
    var result = VisualStateManager.GoToState(this, "WindowsPhone", false);
#endif
```



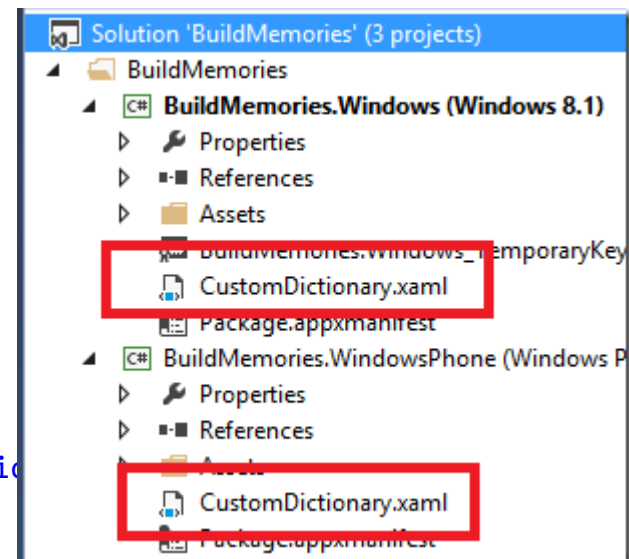
Közös XAML nézetek

2. Platform specifikus „átlátszó” erőforrás fájlok

Nagyon cool 😊

- > 2 erőforrásfájl 1 helyen definiálva

```
<Application
  x:Class="FlickrSearch.App"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:local="using:FlickrSearch">
  <Application.Resources>
    <ResourceDictionary>
      <ResourceDictionary.MergedDictionaries>
        <ResourceDictionary Source="CustomDictionary.xaml" />
      </ResourceDictionary.MergedDictionaries>
    </ResourceDictionary>
  </Application.Resources>
</Application>
```



XAML megosztása

- Konklúzió

Don't think I encourage you to share the maximum percentage of your XAML code : this can be very tricky and difficult to understand what styles, datatemplates and usercontrols are shared or not, and how they relate to each other.

Forrás: msdn blog

Közös alkalmazásfejlesztés

F1-live.hu hírolvasó és SValley

F1-live.hu hírolvasó és SValley

- Universal app
- Shared és nem shared XAML
 - > GridView, Browser, Hub
- Közös logika
 - > RSS letöltése, közös ViewModel
 - > Shared API példa (Geofence)
- Fordítás direktívák
- Speciális esetek (navigáció)
- Erőforrások megosztása, vagy mégse?
 - > Sablonok, vezérlők megjelenítése, színek

Windows Phone Silverlight

- Hova tovább? Mikor jó?
 - > Silverlight XAML – régi megjelenítési motor, performanciában elmarad WinRT-s társától.
- Régi appom van hogyan döntsek?
 - > Meglévő WP7, WP8 alkalmazások módosítása (= túl sok idő alapjaiban újraírni egy komplex programot)
 - Erőforrás / Finanszírozási kihívások
- „Hibrid” API támogatás (pl.: életciklus)
 - > [http://msdn.microsoft.com/en-us/library/windows/apps/dn642084\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windows/apps/dn642084(v=vs.105).aspx)
- Background Task és Background Agent is támogatott



.NET 2015

Rengeteg, de rengeteg újdonság

.NET open source

- 2014 tavasz - .NET Foundation - ASP.NET, EF, Web API, NuGet and the "Roslyn" C# and VB compilers
- 2014 november - **.NET Core Runtime**
 - > CLR, Just-In-Time Compiler (JIT), Garbage Collector (GC), core .NET base class libraries
 - > MIT liszensz
 - Elv: bárki küldhet be módosításokat; ha a .NET team elfogadja van rá esély, hogy bekerül a framework-be.
 - <https://github.com/dotnet/corefx>
 - Egyenlőre: ImmutableCollections, SIMD, XML and MetadataReader (hamarosan jön a többi!)

.NET open source

- MIT JELENT MINDEZ?

- > Teljes támogatás
- > Nyílt forráskód
- > Cross platform

.NET (SERVER) STACK (fordító → runtime → BCL)

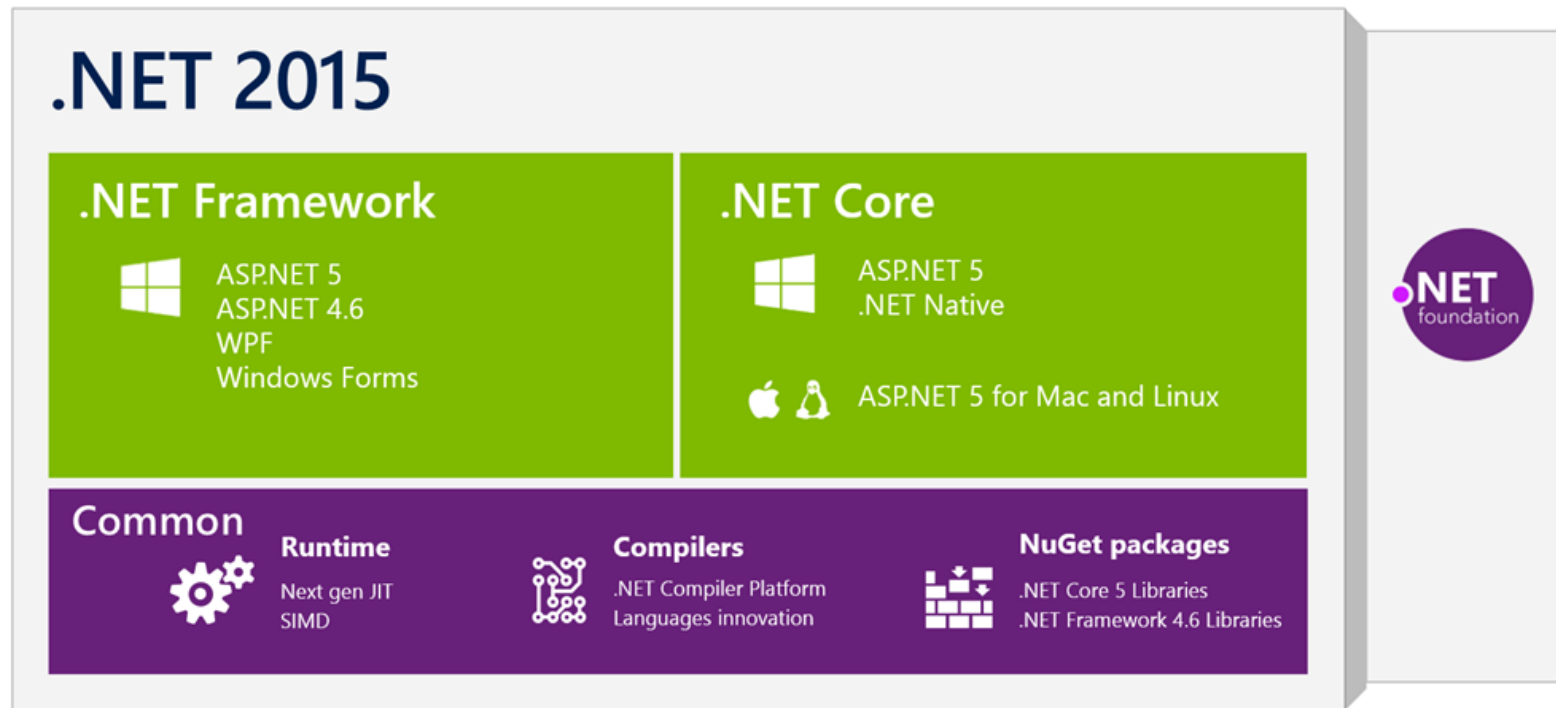


MAC OS, LINUX disztribúciók hivatalos megjelenése
/ASP.NET, SERVER, CLOUD appok fejlesztése/

.NET open source

- Akkor hogy is van ez?

The open .NET ecosystem

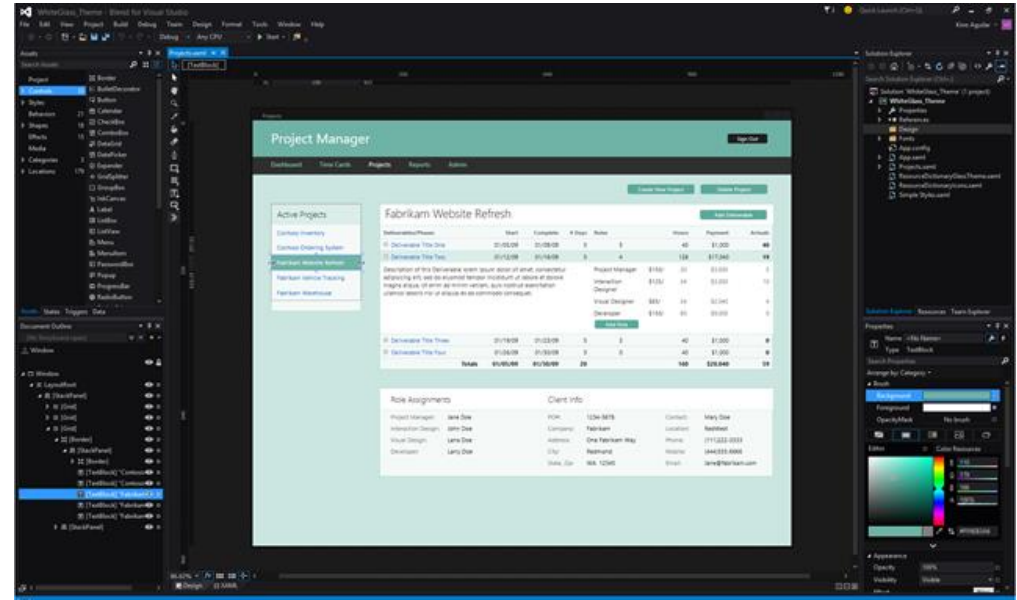


Visual Studio 2015 Preview újdonságok

- Intenzív mobil cross platform támogatás:
 - > Xamarin
 - > C++
 - > Apache Cordova
 - > Android Emulator
- Nyelvi és debugger újdonságok
 - > Await try catch blokkban
 - > Lambda kifejezések debuggolása Watch ablakban
 - > Most először: refactor javaslatok (nem profiler!)
 - > ASP.NET: in memory compilation (bye-bye bin folder)
- Teljes lista: <http://www.visualstudio.com/news/vs2015-preview-vs>

Blend újdonságok

- Visual Studio-val megegyező GUI
 - Pl.: Solution Explorer
- Debugging opciók
- Automatikus reload élmény



C# 6 újdonságok

- Filozófia
 - > Sok apróság
 - > Tisztább kód

Getter-only auto-properties

```
public class Point
{
    public int X { get; set; }
    public int Y { get; set; }
    public Point(int x, int y) { X = x; Y = y; }
    public double Dist
    {
        get { return Math.Sqrt(X * X + Y * Y); }
    }
    public override string ToString()
    {
        return String.Format("({0}, {1})", X, Y);
    }
}
```

Getter-only auto-properties

```
public class Point
{
    public int X { get; }
    public int Y { get; }
    public Point(int x, int y) { X = x; Y = y; }
    public double Dist
    {
        get { return Math.Sqrt(X * X + Y * Y); }
    }
    public override string ToString()
    {
        return String.Format("({0}, {1})", X, Y);
    }
}
```

Getter-only auto-properties

```
public class Point
{
    public int X { get; }
    public int Y { get; }
    public Point(int x, int y) { X = x; Y = y; }
    public double Dist
    {
        get { return Math.Sqrt(X * X + Y * Y); }
    }
    public override string ToString()
    {
        return String.Format("({0}, {1})", X, Y);
    }
}
```


Initializers for auto-properties

```
public class Point
{
    public int X { get; } = 5;
    public int Y { get; } = 7;
    public Point(int x, int y) { X = x; Y = y; }
    public double Dist
    {
        get { return Math.Sqrt(X * X + Y * Y); }
    }
    public override string ToString()
    {
        return String.Format("({0}, {1})", X, Y);
    }
}
```

Using static classes

```
using System.Math;

public class Point
{
    public int X { get; }
    public int Y { get; }
    public Point(int x, int y) { X = x; Y = y; }
    public double Dist
    {
        get { return Math.Sqrt(X * X + Y * Y); }
    }
    public override string ToString()
    {
        return String.Format("({0}, {1})", X, Y);
    }
}
```

Using static classes

```
using System.Math;

public class Point
{
    public int X { get; }
    public int Y { get; }
    public Point(int x, int y) { X = x; Y = y; }
    public double Dist
    {
        get { return Math.Sqrt(X * X + Y * Y); }
    }
    public override string ToString()
    {
        return String.Format("({0}, {1})", X, Y);
    }
}
```

Using static classes

```
using System.Math;

public class Point
{
    public int X { get; }
    public int Y { get; }
    public Point(int x, int y) { X = x; Y = y; }
    public double Dist
    {
        get { return Sqrt(X * X + Y * Y); }
    }
    public override string ToString()
    {
        return String.Format("({0}, {1})", X, Y);
    }
}
```

String interpolation

```
using System.Math;

public class Point
{
    public int X { get; }
    public int Y { get; }
    public Point(int x, int y) { X = x; Y = y; }
    public double Dist
    {
        get { return Sqrt(X * X + Y * Y); }
    }
    public override string ToString()
    {
        return String.Format("({0}, {1})", X, Y);
    }
}
```

String interpolation

```
using System.Math;

public class Point
{
    public int X { get; }
    public int Y { get; }
    public Point(int x, int y) { X = x; Y = y; }
    public double Dist
    {
        get { return Sqrt(X * X + Y * Y); }
    }
    public override string ToString()
    {
        return $"{X}, {Y}";
    }
}
```

Expression-bodied methods

```
using System.Math;

public class Point
{
    public int X { get; }
    public int Y { get; }
    public Point(int x, int y) { X = x; Y = y; }
    public double Dist
    {
        get { return Sqrt(X * X + Y * Y); }
    }
    public override string ToString()
    {
        return $"{X}, {Y}";
    }
}
```

Expression-bodied methods

```
using System.Math;

public class Point
{
    public int X => { return "\{X}, \{Y}"; }
    public int Y
    public Point() => "\{X}, \{Y}"
    public double
    {
        get { return System.Math.Sqrt(X * X + Y * Y); }
    }
    public override string ToString()
    {
        return "\{X}, \{Y}";
    }
}
```


Expression-bodied methods

```
using System.Math;

public class Point
{
    public int X { get; }
    public int Y { get; }
    public Point(int x, int y) { X = x; Y = y; }
    public double Dist
    {
        get { return Sqrt(X * X + Y * Y); }
    }
    public override string ToString() => $"{X}, {Y}";
}
```

Expression-bodied properties

```
using System.Math;

public class Point
{
    public int X { get; }
    public int Y { get; }
    public Point(int x, int y) { X = x; Y = y; }
    public double Dist
    {
        get { return Sqrt(X * X + Y * Y); }
    }
    public override string ToString() => "{X}, {Y}";
}
```

Expression-bodied properties

```
using System.Math;

public class Point
{
    public int X { get; }
    public int Y { get; }
    public Point(int x, int y) { X = x; Y = y; }
    public double Dist => Sqrt(X * X + Y * Y);
    public override string ToString() => $"{X}, {Y}";
}
```

Expression-bodied properties

```
using System.Math;

public class Point
{
    public int X { get; }
    public int Y { get; }
    public Point(int x, int y) { X = x; Y = y; }
    public double Dist => Sqrt(X * X + Y * Y);
    public override string ToString() => "{X}, {Y}";
}
```

Index initializers

```
public class Point
{
    public int X { get; }
    public int Y { get; }
    ...
    public JObject ToJson()
    {
        var result = new JObject();
        result["x"] = X;
        result["y"] = Y;
        return result;
    }
}
```

Index initializers

```
public class Point
{
    public int X { get; }
    public int Y { get; }
    ...
    public JObject ToJson()
    {
        var result = new JObject() { ["x"] = X, ["y"] = Y };
        return result;
    }
}
```

Index initializers

```
public class Point
{
    public int X { get; }
    public int Y { get; }
    ...
    public JObject ToJson()
    {
        return new JObject() { ["x"] = X, ["y"] = Y };
    }
}
```

Index initializers

```
public class Point
{
    public int X { get; }
    public int Y { get; }
    ...
    public JObject ToJson() =>
        new JObject() { ["x"] = X, ["y"] = Y };
}
```


Null-conditional operators

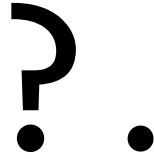
```
public static Point FromJson(JObject json)
{
    if (json != null &&
        json["x"] != null &&
        json["x"].Type == JTokenType.Integer &&
        json["y"] != null &&
        json["y"].Type == JTokenType.Integer)
    {
        return new Point((int)json["x"], (int)json["y"]);
    }
    return null;
}
```

Null-conditional operators

```
public static Point FromJson(JObject json)
{
    if (json != null &&
        json["x"]?.Type == JTokenType.Integer &&
        json["y"]?.Type == JTokenType.Integer)
    {
        return new Point((int)json["x"], (int)json["y"]);
    }
    return null;
}
```

Null-conditional operators

```
public static Point FromJson(JObject json)
{
    if (json != null &&
        json["x"]?.Type == JTokenType.Integer &&
        json["y"]?.Type == JTokenType.Integer)
    {
        return new Point((int)json["x"], (int)json["y"]);
    }
    return null;
}
```



Null-conditional operators

```
public static Point FromJson(JObject json)
{
    if (json != null &&
        json["x"]?.Type == JTokenType.Integer &&
        json["y"]?.Type == JTokenType.Integer)
    {
        return new Point((int)json["x"], (int)json["y"]);
    }
    return null;
}
```

Null-conditional operators

```
public static Point FromJson(JObject json)
{
    if (json?["x"]?.Type == JTokenType.Integer &&
        json?["y"]?.Type == JTokenType.Integer)
    {
        return new Point((int)json["x"], (int)json["y"]);
    }
    return null;
}
```

Null-conditional operators

```
OnChanged(this, args);
```

Null-conditional operators

```
if (OnChanged != null)
{
    OnChanged(this, args);
}
```

Null-conditional operators

```
{  
    var onChanged = OnChanged;  
    if (onChanged != null)  
    {  
        onChanged(this, args);  
    }  
}
```


Null-conditional operators

```
OnChanged?.Invoke(this, args);
```

The nameof operator

```
public Point Add(Point point)
{
    if (point == null)
    {
        throw new ArgumentNullException("point");
    }
}
```

The nameof operator

```
public Point Add(Point other)
{
    if (other == null)
    {
        throw new ArgumentNullException("point");
    }
}
```

The nameof operator

```
public Point Add(Point point)
{
    if (point == null)
    {
        throw new ArgumentNullException(nameof(point));
    }
}
```

The nameof operator

```
public Point Add(Point other)
{
    if (other == null)
    {
        throw new ArgumentNullException(nameof(other));
    }
}
```

Exception filters

```
try
{
    ...
}
catch (ConfigurationException e)
{

}
finally
{

}
```

Exception filters

```
try
{
    ...
}
catch (ConfigurationException e) if (e.IsSevere)
{

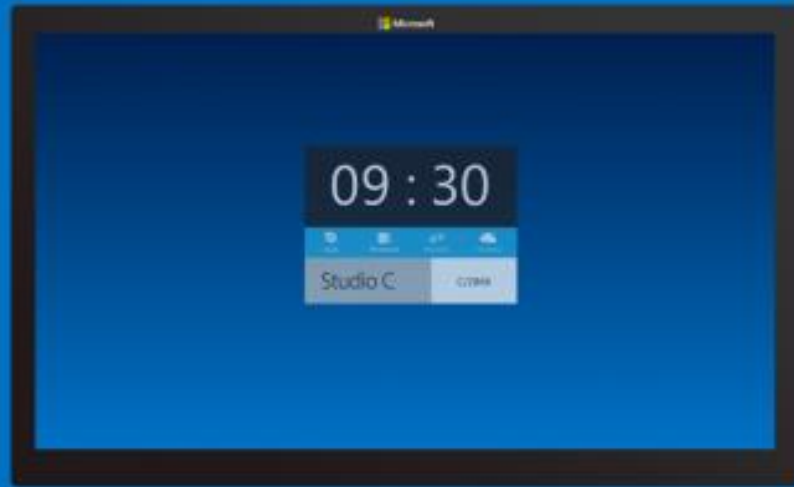
}
finally
{

}
```

Await in catch and finally

```
try
{
    ...
}
catch (ConfigurationException e) if (e.IsSevere)
{
    await LogAsync(e);
}
finally
{
    await CloseAsync();
}
```


WINDOWS 10



Windows 10



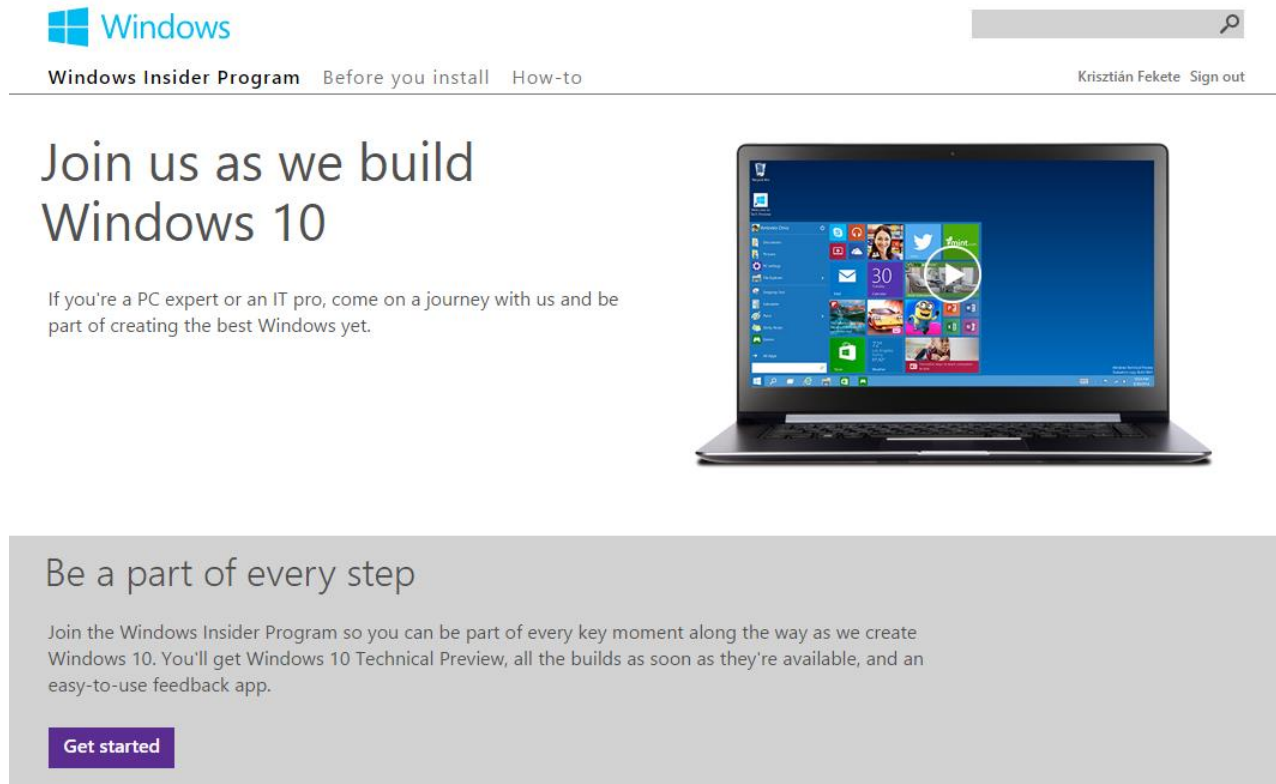
One product family

One platform

One store

Windows 10 letöltés

- Windows Insider Program (feedback-Insider Hub app)
> <http://windows.microsoft.com/en-us/windows/preview>




The screenshot shows the Windows Insider Program website. At the top, there's a navigation bar with the Windows logo, the text "Windows", and a search icon. Below this, there's a header with "Windows Insider Program" and links for "Before you install" and "How-to". On the right side of the header, the user "Krisztián Fekete" is logged in, with a "Sign out" link. The main content area has the heading "Join us as we build Windows 10" and a subheading "If you're a PC expert or an IT pro, come on a journey with us and be part of creating the best Windows yet." Below this text is an image of a laptop displaying the Windows 10 desktop. At the bottom, there's a grey box with the heading "Be a part of every step" and a paragraph: "Join the Windows Insider Program so you can be part of every key moment along the way as we create Windows 10. You'll get Windows 10 Technical Preview, all the builds as soon as they're available, and an easy-to-use feedback app." A purple "Get started" button is located at the bottom left of this grey box.

Windows Insider Program Before you install How-to Krisztián Fekete Sign out

Join us as we build Windows 10

If you're a PC expert or an IT pro, come on a journey with us and be part of creating the best Windows yet.



Be a part of every step

Join the Windows Insider Program so you can be part of every key moment along the way as we create Windows 10. You'll get Windows 10 Technical Preview, all the builds as soon as they're available, and an easy-to-use feedback app.

[Get started](#)

Windows 10 letöltés

- Már 1 millióan regisztráltak
- „Slow ring users”: csak a stabil verziót kapják meg
- „Fast ring users”: a nem stabil verziókat is megkapjuk update-ként (kb. 10% most)
- PC Settings-ben beállítható
 - Klasszikus Control Panel valószínűleg kikerül a végleges verzióból

Windows 10

- Miért nem 9?
 - > version.StartsWith("Windows 9")
 - > Sok újdonság: "It came and it went".
- Kernel verziószám: 10 (Korábban 6.x)

*Developers preparing for Windows 10 should note that the Windows NT value in the **UA string will change from 6.4 to 10.0 with the new release.** If you have code that depends on the version number, we advise that you update to allow for the new value.*

- Konvergencia befejeződése
 - > Telefon, Tablet, Desktop, XBOX
 - one Windows Store for all Windows devices
- Várható megjelenés: 2015 nyár
 - > Most: Technical Preview (Build: 9879)
 - > Consumer Preview: 2015 január

Windows 10

- Dedikált Store élmény cégek számára
- Windows 10: minden WP8 / 8.1 eszköz is frissíthető lesz

Windows 10 újdonságok

- START menü
 - > Átméretezhető
 - > Eredeti Start screen visszaállítható



Windows 10 újdonságok

- Command line
- CTRL+V / CTRL + C!!!



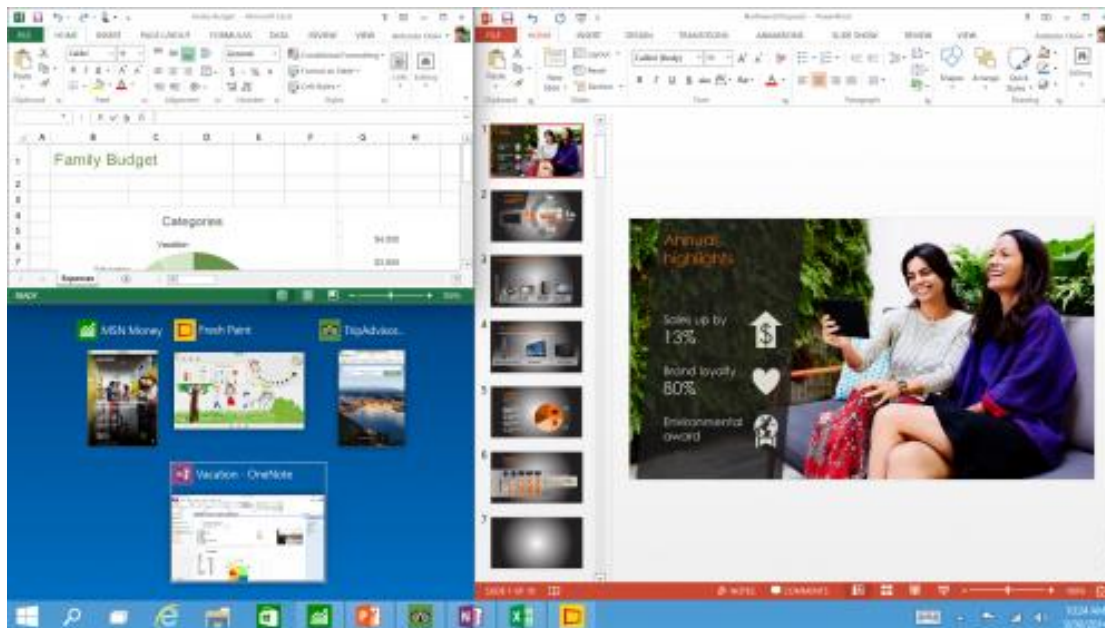
A screenshot of a Windows Command Prompt window. The title bar shows the path `C:\Windows\system32\cmd.exe`. The window content displays the following text:

```
Microsoft Windows [Version 6.1.7601]  
Copyright (c) 2009 Microsoft Corporation. All rights reserved.  
C:\Users\w7vm>^U^U^U^U
```

The command `^U^U^U^U` represents the keyboard sequence for pasting (Ctrl+V) in a standard Windows command prompt environment.

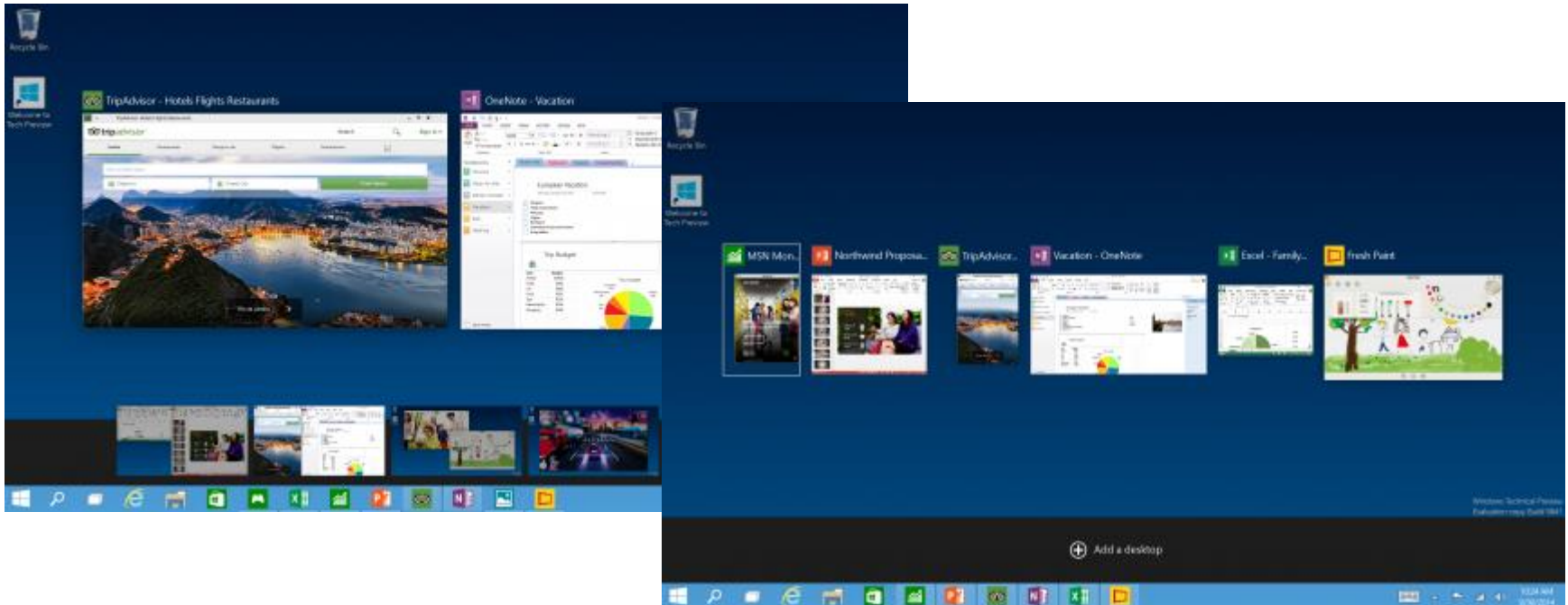
Windows 10 újdonságok

- Snap újdonságok
 - > Már 4 alkalmazást snap-elhetünk egyszerre
 - > A rendszer **javaslatokat** tesz nekünk a következő snap-re



Windows 10 újdonságok

- Új Task viewer és Virtual Desktops
 - > Egyszerre több asztal (pl.: munkára, otthonra stb.)
 - > Váltás: Windows + CTRL + [Left|Right] Key

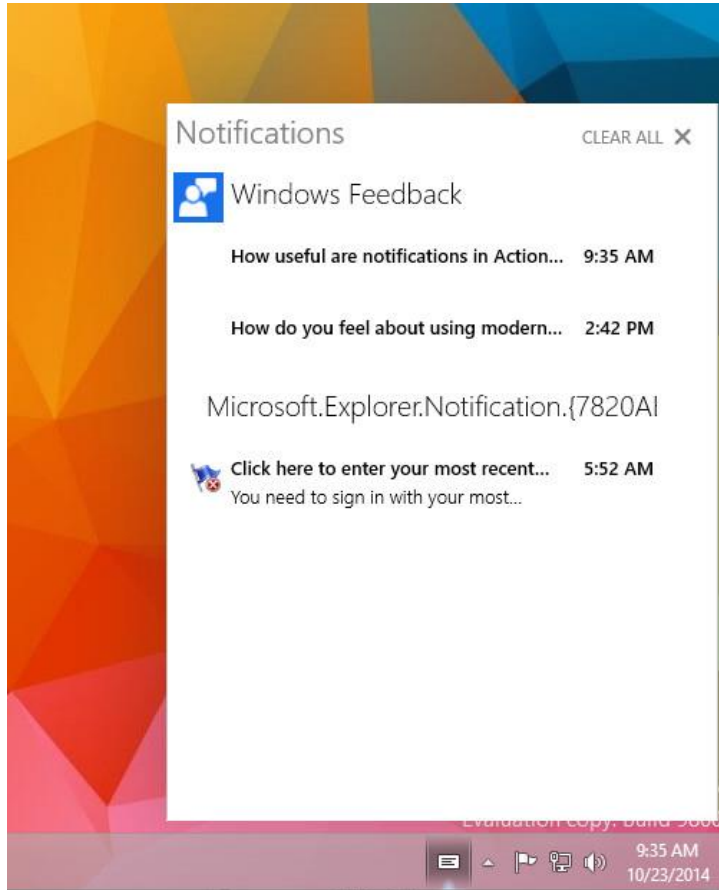


Windows 10 újdonságok

- Átméretezhető tablet alkalmazások
- Globális keresés
- Tablet appok már nem csak Taskbar ikont kapnak hanem a Desktop-ra is kirakhatók
- Explorer:
 - > Recent Files
 - > Share
 - > Favorites
- UI finomítások
 - > Árnyékok, Flat ikonok stb.
- Desktop élmény hangsúlyosabb (pl.: WiFi választás)

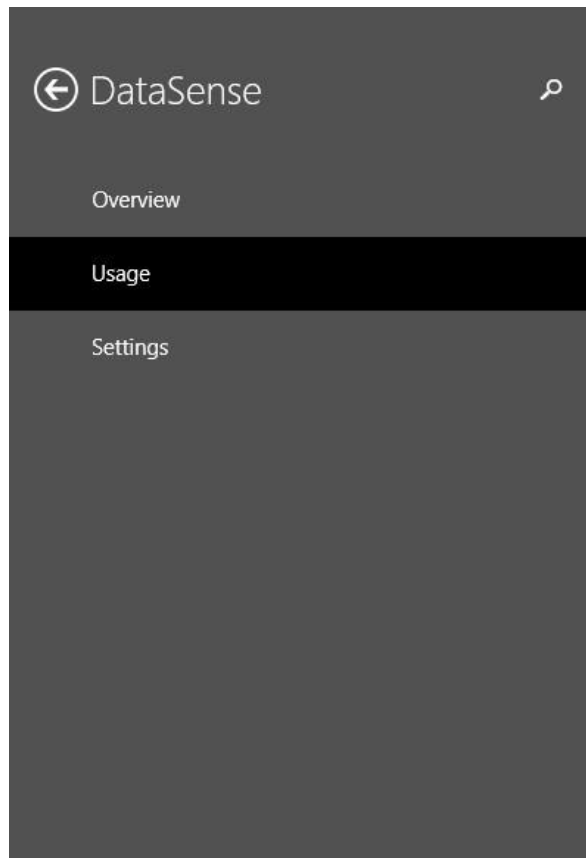
Windows 10 újdonságok

- Action (Notification) Center



Windows 10 újdonságok

- DataSense



Usage

Since: 9/21/2014
Total usage



■ Wi-Fi: 0 bytes
■ Cellular: 0 bytes



System Services

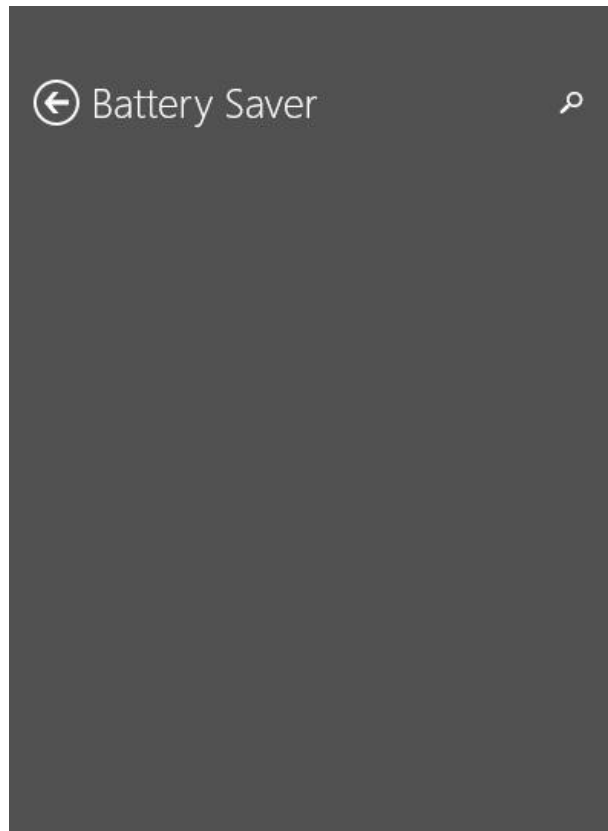
0 bytes

0 bytes



Windows 10 újdonságok

- Battery Saver



Battery Saver

Conserves battery by limiting background activity and adjusting hardware settings. Battery Saver will turn on automatically if you like, but you can also turn it on manually.

When battery saver mode is active, this symbol appears next to the battery icon. (Only active when battery is discharging)

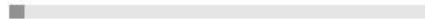
Right now

Off



Automatic Rules

☒ Turn Battery Saver on when my battery gets below this percentage

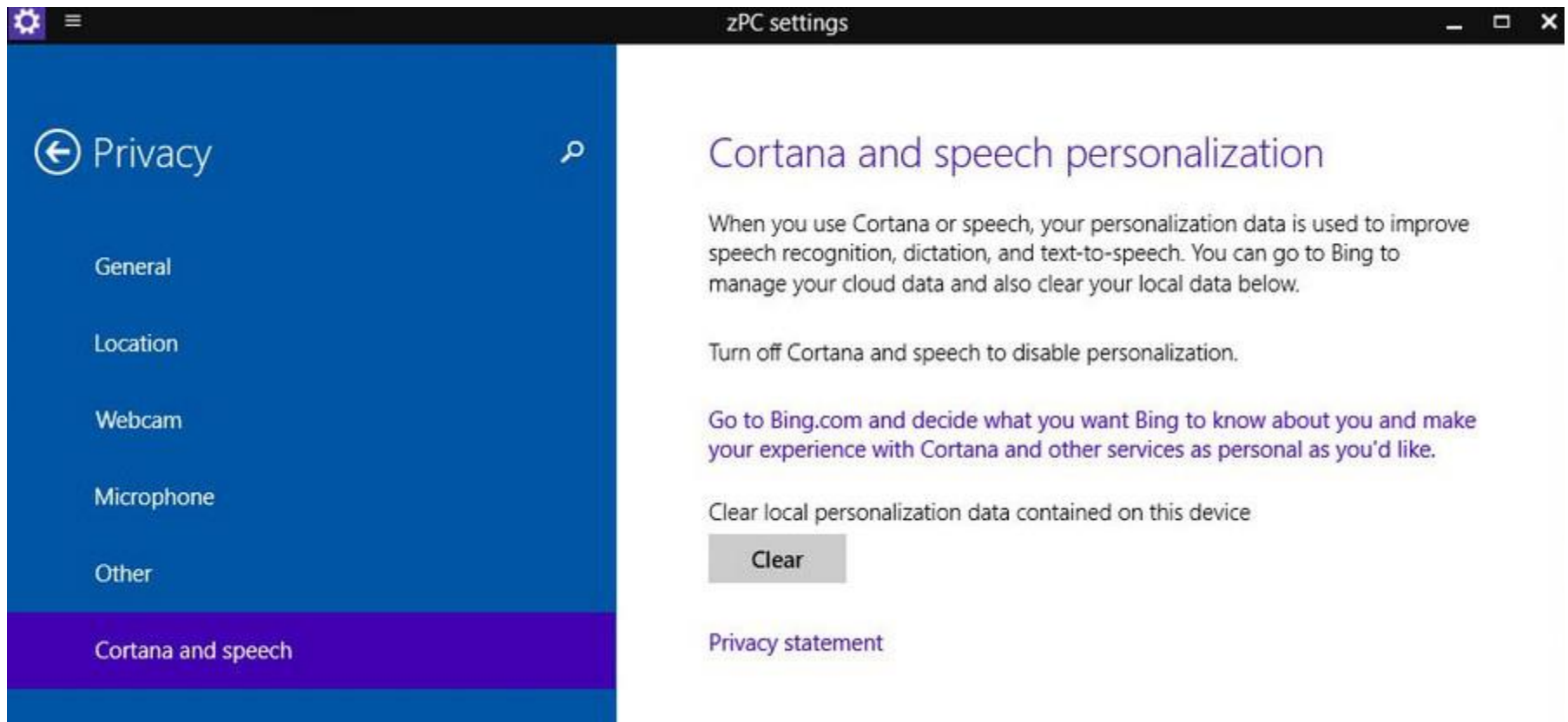


Windows 10 újdonságok

- Trackpad gesztusok:
 - > Three-fingers up: Task View
 - > Three-fingers down: Show Desktop
 - > Three-fingers flick to left or right: Switch between open apps
 - > Three-fingers move left or right: Alt-Tab pops up allowing you to highlight and select the app you want
 - > Three-fingers tap: Search
- <https://www.youtube.com/watch?v=ZHg6JL5sdaI>

Windows 10 újdonságok

- Cortana
 - > zPC settings



Windows 10 újdonságok

- Continuum
 - > Desktop élmény automatikus átváltozása Tablet élménnyé (laptop / tablet mode)
 - > https://www.youtube.com/watch?v=F_O-LrGL-YQ



Windows 10 és az Universal appok

- Sajnos fejlesztési szempontból még nincsenek hírek a Windows 10-ről
- Amire számítani lehet: több eszköz támogatása a telefon és a tablet eszközök mellett (Desktop, Konzol?)
- Várható újdonságok:
 - > Build 2015
 - Április 29 – Május 1 (San Francisco USA)
 - > <http://www.buildwindows.com/>

Köszönöm a figyelmet!

Fekete.Krisztian@aut.bme.hu