Oliver Sturm

BASTA!

.NET, WINDOWS, VISUAL STUDIO

C# in the Modern World

Oliver Sturm (@olivers)

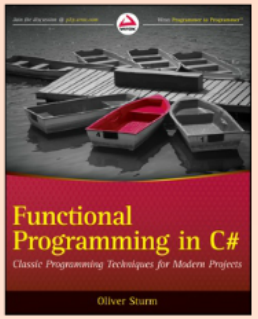
- Consultant, Trainer, Author
- Associate Consultant at thinktecture
- Course Author at Pluralsight

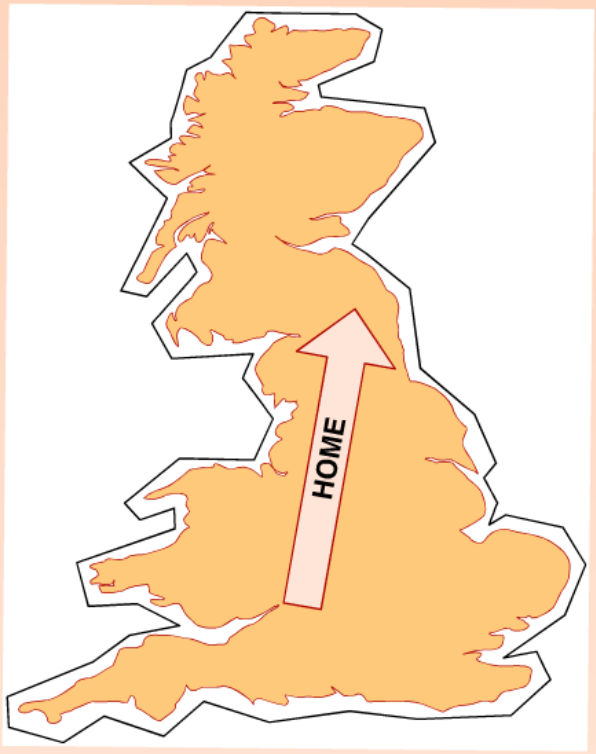
thinktecture

pluralsight
see what you can learn

DevExpress™ Expert

Microsoft® Most Valuable Professional





Contact

oliver@oliversturm.com

Services

http://www.oliversturm.com

Agenda

- What's new in C# 5.0?
- The new target: Metro
- What about the other languages?

What's new in C# 5.0

1) await
2) async

... hm, that's it. Pretty much.

DEMO

Targeting Metro (or Modern UI, Windows Store Applications, whatever)

- Windows Runtime (WinRT) is the platform
- For C#, the CLR provides additional runtime services
- A restricted .NET framework profile is in use
- Windows Store Applications in C# use a XAML based UI model

DEMO

What about my existing code?

- Several different types of library projects are supported in VS 2012
- Either create WinRT libraries for your code or stay Portable
- Building existing code in the new environment depends on API availability

DEMO

Language options

- C#, VB and other .NET languages target WinRT through the CLR
- C++/CX provides a projection based on the .winmd metadata
- C++/COM/WRL (Windows Runtime C++ Template Library, somewhat similar to ATL) goes through COM "directly"
- JavaScript has a .winmd based projection

C#, VB, and other .NET languages

- Elegant languages: async/await, LINQ, dynamic, C#/F# are very "modern"
- Benefits of established platform diminished by restricted .NET profile
- Potential for mixed target platforms: e.g. full application on full .NET, restricted/specialized version for Metro/Windows Store

C++(/CX)

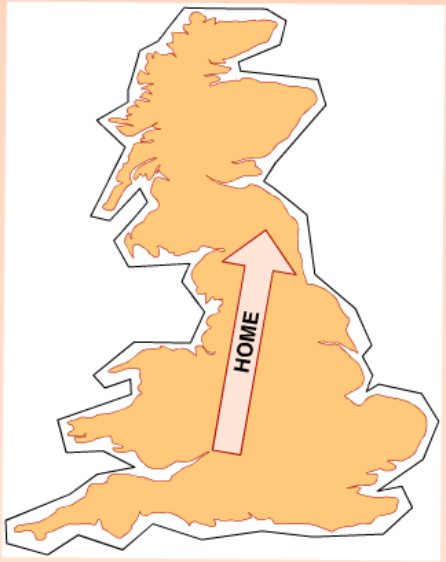
- Well, it's C++, basically
- It's not as bad as MFC C++. But still...
- Most flexible option:
  - Native .exe
  - Target other platforms that do C++
  - Templates for DirectX 2D and 3D integration
  - C++ AMP (Accelerated Massive Parallelism) for GPU computations through DirectX 11
- Slow compilation, huge projects, ...

JavaScript

- Use HTML/CSS instead of XAML
- Targeting of a whole different set of platforms
- Enormously flexible deployment scenarios
- For some developers, a very big step to make
- Some technical restrictions
  - Can't implement WinRT interfaces
  - Important? Not sure
- My guess: aspect of interoperability outside WinRT important for many

DEMO

Thank you for watching!



Contact

oliver@oliversturm.com

Services

http://www.oliversturm.com