

Creating Reusable Unit Tests Across Platforms

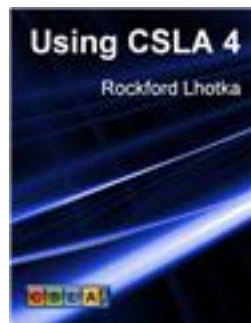
Rockford Lhotka
Magenic
rocky@lhotka.net

Magenic

Custom solutions that fit. Guaranteed.



Microsoft
Regional Director
PROGRAM



Background

CSLA .NET

.NET 2001 - now

Silverlight 2007 – now

Windows Phone 2010 – now

Mono 2011 – now

WinRT <prototype>

Tests

nunit -> mstest

UnitDriven

Silverlight

WP7

Android

UnitDriven

- <http://unitdriven.codeplex.com>
- Consistent tests across platforms
- Integrates with
 - nunit
 - mstest
- Runners for
 - Silverlight
 - Android

Other Tools

Silverlight	Microsoft Unit Test Framework
-------------	-------------------------------

	nunit
--	-------

	StatLight
--	-----------

Windows Phone	Microsoft Unit Test Framework (tweaked)
---------------	-----------------------------------------

	nunit
--	-------

mono for Android / monotouch	???
------------------------------	-----

Unit and Integration Testing

Developer authored tests

Tests at a deep code level

Unit tests = test isolated methods

Integration tests = test several methods

The “Old World”

Code and tests run on one platform

N-tier apps run on Windows
client/server

Most code synchronous

Modern World

Microsoft

- Windows/.NET
- Silverlight
- Windows Phone
- WinRT

Other

- Mono
- Mono for Android
- Mono for iOS

Architectures

N-tier

- App code spans multiple layers/tiers
- Tiers may be on various platforms

SOA

- System integrates numerous apps
- Apps span multiple layers/tiers
- Apps and tiers on various platforms

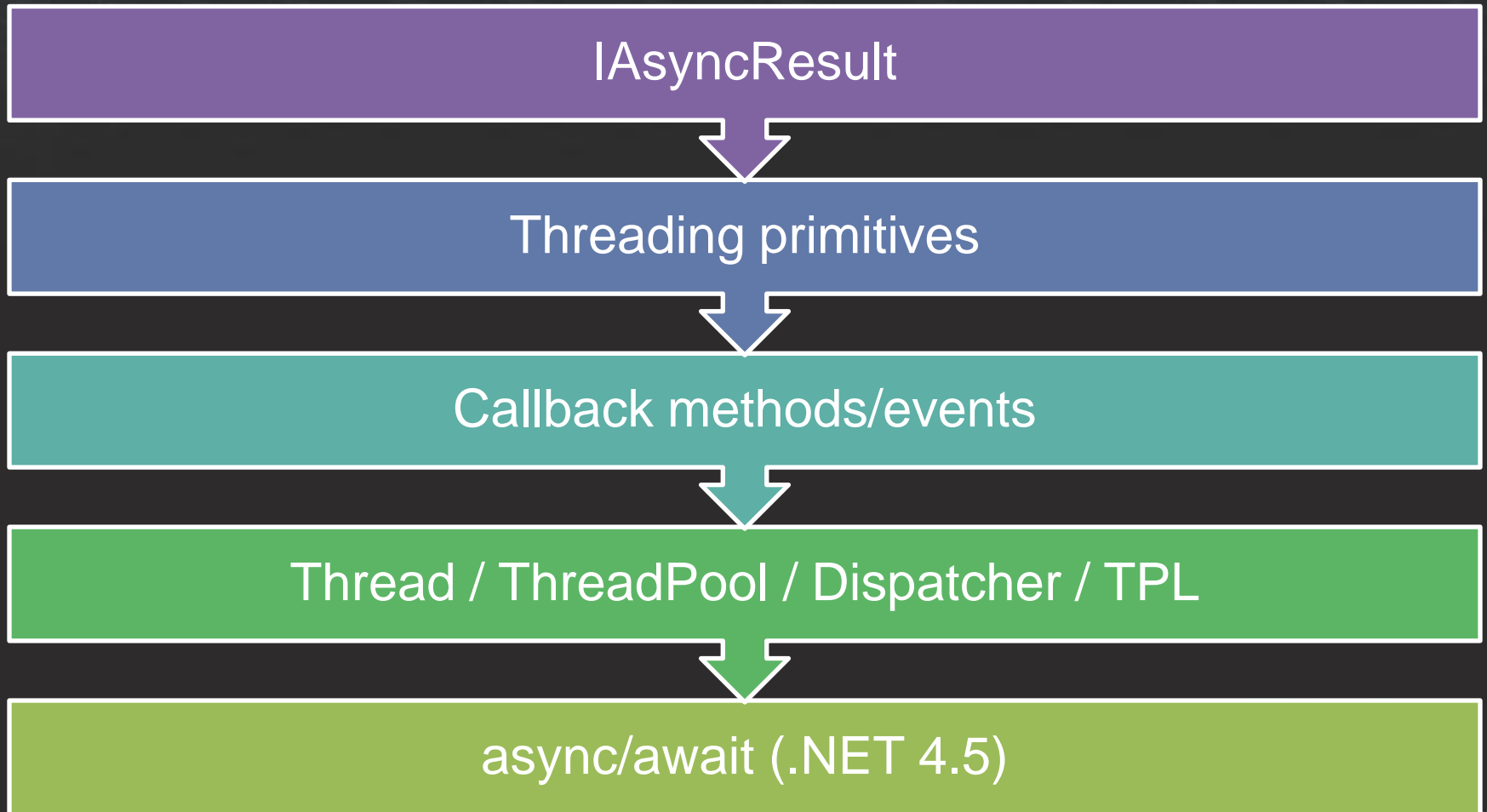
Synchronous vs Asynchronous

Most code still synchronous

Server access asynchronous

Growing prevalence of async

Asynchronous Code Models



Testing Across Platforms

Why?

Same code behaves differently

Code differences between platforms

Goal

Reuse same tests across platforms

Requirements

Consistent test framework

Per-platform test runners



Complications

Platform differences

- Threading behaviors
- Reflection behaviors
- .NET capabilities (e.g. lambda expressions)
- BCL differences

Runner implementation

- Device screen sizes
- Getting results off devices
- Automating tests in emulators

Testing Across Tiers

Why?

Some behaviors can only be tested if they span AppDomain/Thread boundaries

Goal

Minimize complexity of n-tier tests

Requirements

Test configuration

Managing async tests

Complications

Spinning up servers/services for tests

Test configuration issues

Network security

Simulating network via AppDomain/Thread

Testing Asynchronous Code

Why?

Increasing use of async in coding

Hard requirements for async (Silverlight)

Goal

Reuse same tests across platforms

Minimize complexity

Requirements

Deal with different platform
threading models

Complications

Thread affinity

- Simulating the “UI thread”

Platform threading differences

- WinRT vs .NET

Global exception handling

- Not available on Android

Language differences

- async/await vs other techniques

Demo



Your Feedback is Important

Please fill out a session evaluation form
drop it off at the conference registration
desk.

Thank you!

Magenic

Custom solutions that fit. Guaranteed.



Microsoft
Regional Director
PROGRAM

