

xUnit.net v2: Design and Extensibility

James Newkirk, Brad Wilson

<https://github.com/xunit>

@jamesnewkirk @bradwilson @xunit

Why Build xUnit.net?

- **Flexibility:** static and private methods
- **Reduce Friction:** fewer attributes
- **Safety:** create a new instance for every test
- **Be explicit:** no control flow in attributes
- **Runners:** be everywhere the developer is
- **Consistency:** Prefer the language & framework
- **Extensibility:** not an afterthought
- **TDD first:** built with and for TDD

Unexpected Popularity

- We built what we wanted, not what we thought our “customers” would want
- Internal adoption at Microsoft (Roslyn, ASP.NET, Entity Framework, CodePlex, etc.)
- Lots of people built their own “test frameworks” to leverage us (and our runners)
- Half a million downloads (plus whatever the other frameworks have)

Why Build v2?

How stupid is it to start over when you're popular?
Haven't you heard of "second system syndrome"?

(Spoiler: We can be pretty stupid when we try.)

Evolution of the .NET Framework

- No lambdas
- No LINQ
- Worst. Async. Programming. Model. Ever.

4.5 is a no-brainer. Time to be cutting edge again!

Parallelization

- Growing test suite sizes
 - 10K (or 100K) tests is a lot of aggregate time
 - Most developers have 4-8x as many cores today
- Like randomization, helps find subtle test bugs
- Opportunities:
 - Parallelize tests against each other
 - Parallelize test assemblies against each other

Test Collections

- Something between “test class” and “assembly”
- Group tests together to share long-lived fixtures
- Become the parallelization boundary
- By default, each class is a collection
 - Add [Collection] to test classes
 - Optional collection class, decorated with [TestCollection]

Code Rot

- Years of execution pipeline tweaks
- We hit the wall for improvement without breaking version-independent runner support
- Extensibility points were valuable but limited
- 3rd party runners stressed the framework
- Hacked support for things like async

Community Feedback

- Move [Theory] into the main line
- Make Assert easy to remove, replace, extend
- Support for PCL test libraries
- Support for device runners (Windows 8, Windows Phone 8, iPhone, Android)
- Support for 3rd party runners without hacks (Resharper, Code Rush, NCrunch)

v2 Extensibility

Half a decade of developer requests

Extensibility: Assert

- Lots of options
 - Use our Assert class as-is, binary reference
 - Bring in our Assert class as extensible source code
 - Bring in 3rd party assertions (like Shouldly)
 - Create your own 3rd party assertions, using ours

Sample: AssertExtensions

Extensibility: Before/After

- Run code before & after each test runs
- Intended for cross-cutting concerns
 - Access to the method, but not the test class instance
- Customize:
 - Derive from `BeforeAfterTestAttribute`
 - *Applies to: Test class, Test method*

Samples: `AssumeIdentity`, `UseCulture`

Extensibility: Class Fixtures

- Run code before & after all tests in test class
- Customize:
 - Write a class (before = ctor, after = IDisposable.Dispose)
 - Decorate class with `IClassFixture<T>`
 - Optionally accept fixture via constructor injection
 - *Applies to: Test collection, Test class*

Sample: `ClassFixtureExample`

Extensibility: Collection Fixtures

- Run code before & after all tests in test collection
- Customize:
 - Write a class (before = ctor, after = IDisposable.Dispose)
 - Decorate class with ICollectionFixture<T>
 - Optionally accept fixture via constructor injection
 - *Applies to: Test collection*

Sample: CollectionFixtureExample

Extensibility: Theory Data

- Customize:
 - Derive from DataAttribute
 - Default discoverer behavior calls the attribute to get the data (no rich discovery for source-based runners like R#)
 - Optional class which implements ITheoryDataDiscoverer
 - Decorate custom attribute with [DataDiscoverer]
 - *Applies to: Test method*

Sample: ExcelDataExample

Extensibility: Collections / Parallel

- Built in behaviors:
 - Collection per test class (default) or assembly
 - Parallelization (on by default)
 - Maximum threads (*cores x 4* by default)
- Customize:
 - Add [CollectionBehavior]
 - Class that implements `IXunitTestCollectionFactory`
 - Constructor takes `ITestAssembly`
 - Applies to: Assembly

Extensibility: Test Ordering

- Default order is random (but repeatable)
- Customize:
 - Class that implements `ITestCaseOrderer`
 - Apply with `[TestCaseOrderer]`
 - *Applies to: Assembly, Test collection, Test class*

Sample: `TestOrderExamples`

Extensibility: Traits

- Default traits are done with [Trait]
- Customize:
 - Custom attribute that implements ITraitAttribute
 - Class that implements ITraitDiscoverer
 - Add [TraitDiscoverer] to the custom attribute
 - *Applies to: Test class, Test method*

Sample: TraitExtensibility

Extensibility: FactAttribute

- “What does it mean to be a test?”
- Customize:
 - Derive from FactAttribute
 - Class that implements IXunitTestCaseDiscoverer
 - Decorate attribute with [XunitTestCaseDiscoverer]
 - *Applies to: Test method*

Sample: The source for TheoryAttribute 😊

Extensibility: Test frameworks

- “What does it mean to find and run tests?”
- Customize:
 - Implement ITestFramework
 - Implement ITestFrameworkDiscoverer
 - Implement ITestFrameworkExecutor
 - Apply [TestFramework] attribute
 - *Applies to: Assembly*

Sample: SpecificationExamples

Extensibility: Runners

- Version independent runner libraries
 - `xunit.runner.utility.net35` (supports v1 and v2)
 - `xunit.runner.utility.platform` (support v2 only)
 - Windows 8 (only runs in Visual Studio)
 - Windows Phone 8 (only runs in Visual Studio)
 - iOS (via Xamarin)
 - Android (via Xamarin)

Sample: Source to the MSBuild runner 😊

Contribution

- OuterCurve Foundation project
- Project hosted on GitHub
- There are lots of ways to help:
 - Spread the word
 - Report bugs & feature requests
 - Enhance samples & documentation
 - Pull requests

GitHub

- Core framework and runners:
<https://github.com/xunit/xunit>
- Samples:
<https://github.com/xunit/samples>
- Android/iOS runners:
<https://github.com/xunit/xamarin.xunit>
- Resharper runner:
<https://github.com/xunit/resharper-xunit>

Thank You

We're hiring!

<http://centurylinkcloud.com/careers>

Slides:

<http://bit.ly/BradsTalks>