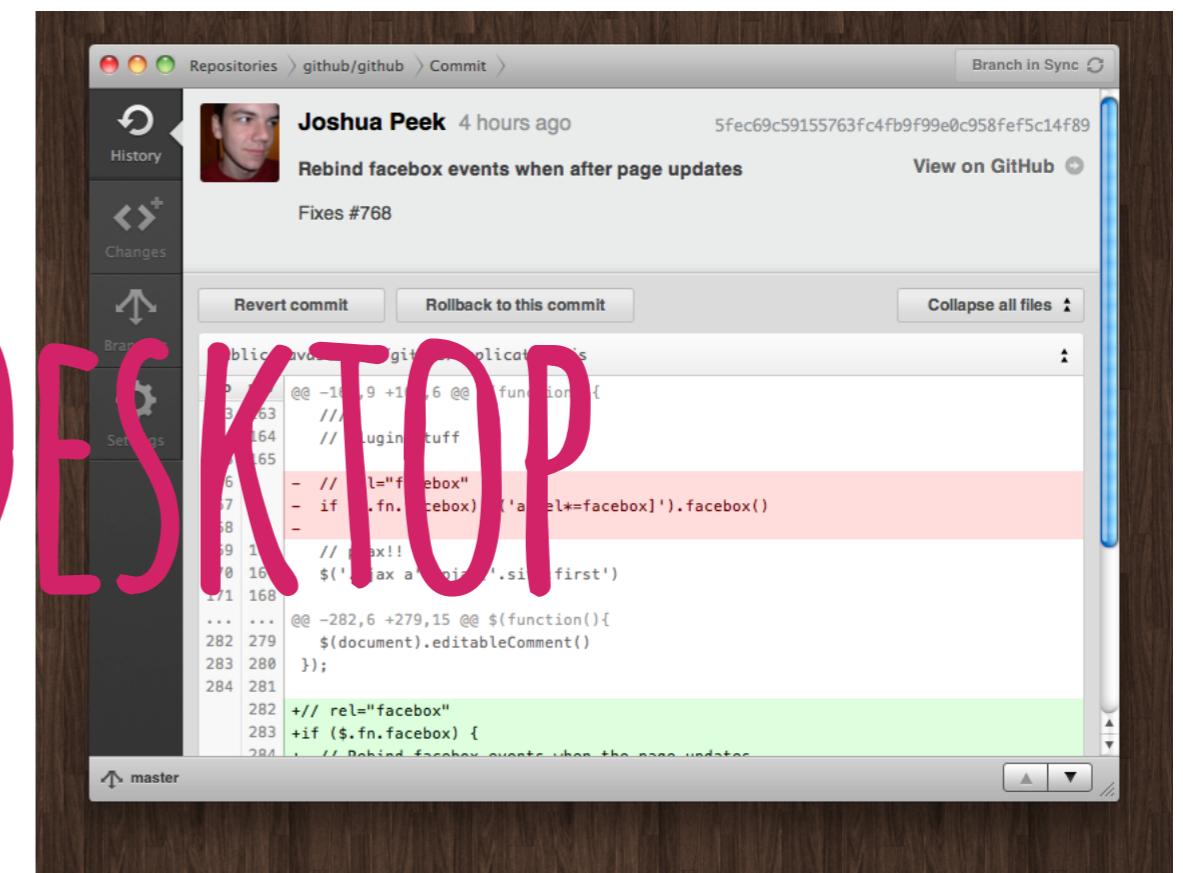
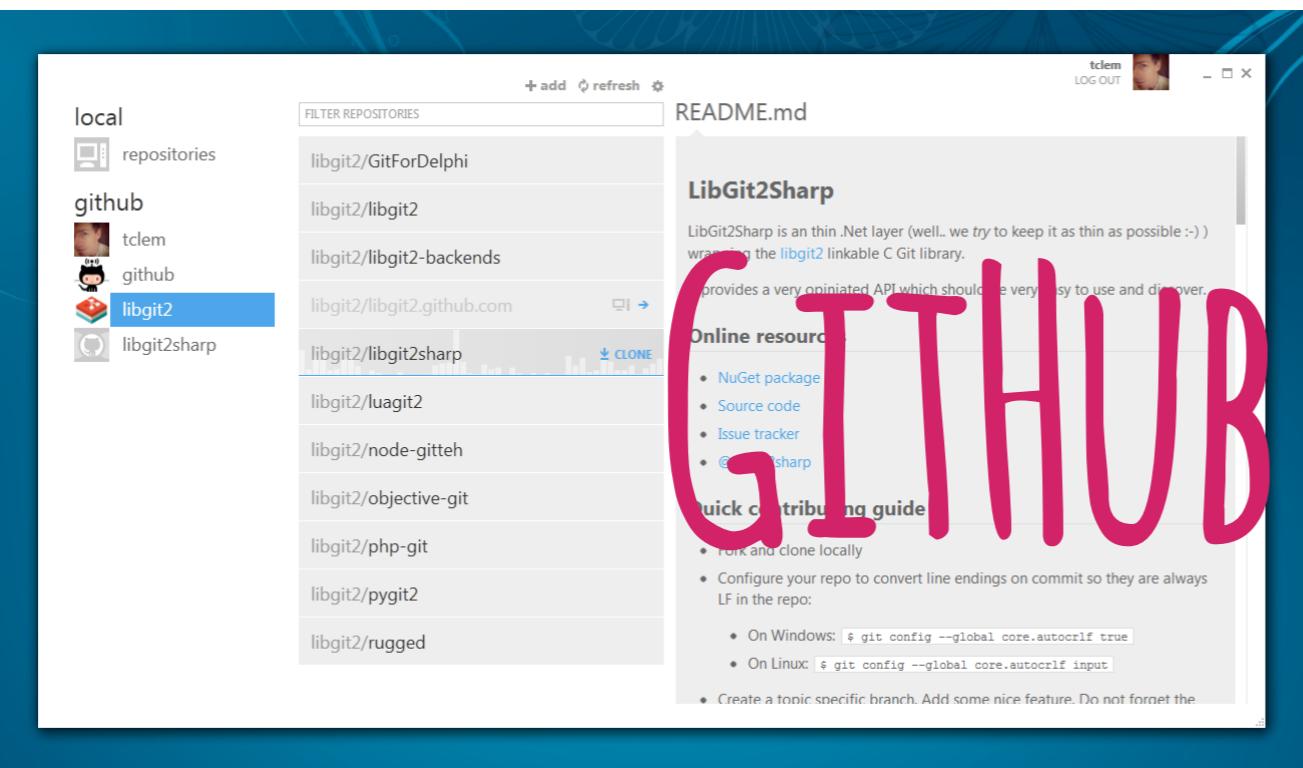


# FRP IN PRACTICE: TAKING A LOOK AT REACTIVE[UI/COCOA]

WHY?

WE CAN NO LONGER WRITE  
SYNCHRONOUS SOFTWARE

– Bob Dylan, quoting Abraham Lincoln



# REACTIVE EXTENSIONS: ASYNC DATA FLOW FRP

# Rx PRIMITIVES

Creating Observables:  
`Observable.Return(42)`

Transforming Observables:  
`something.Select(x => x * 5)`

Changing Contexts:  
`something.ObserveOn(RxApp.MainThreadScheduler)`

# REACTIVEUI: RX + YOUR FAVORITE UI TOOLKIT

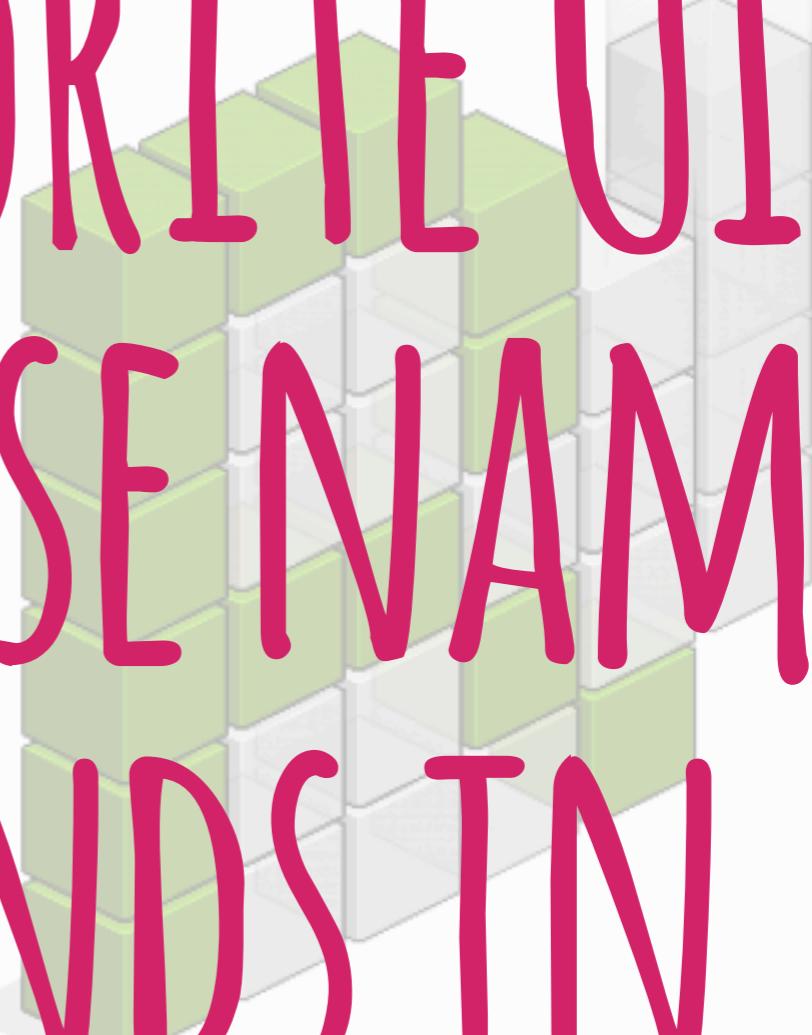


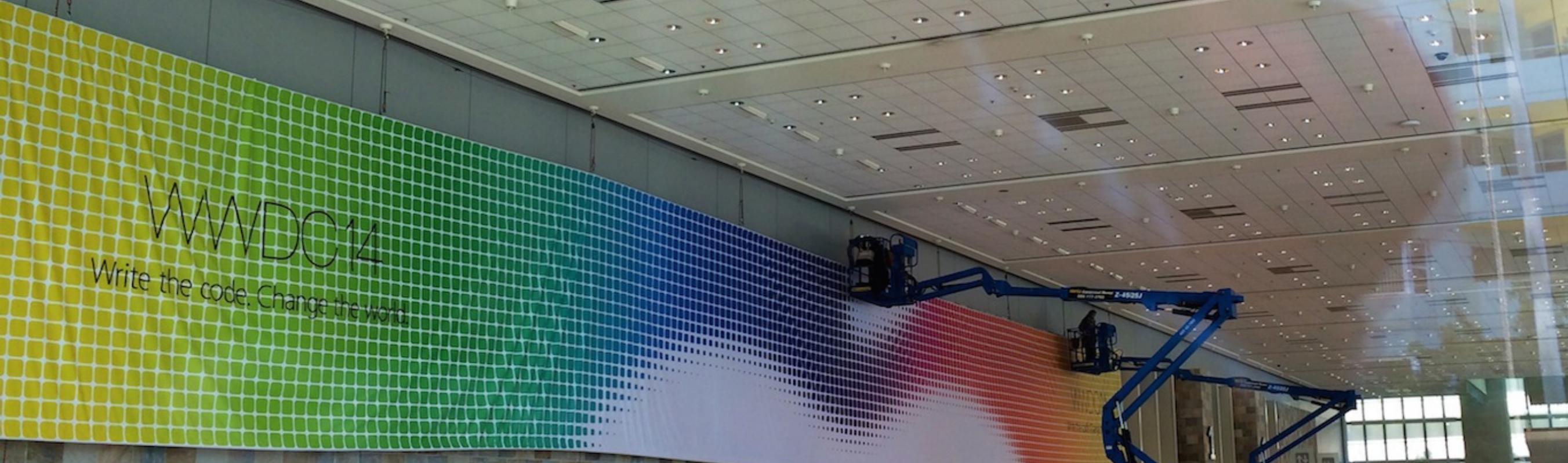
REACTIVECOCOA: Objective-C

+ YOUR FAVORITE UI  
TOOLKIT WHOSE NAME

LITERALLY ENDS IN

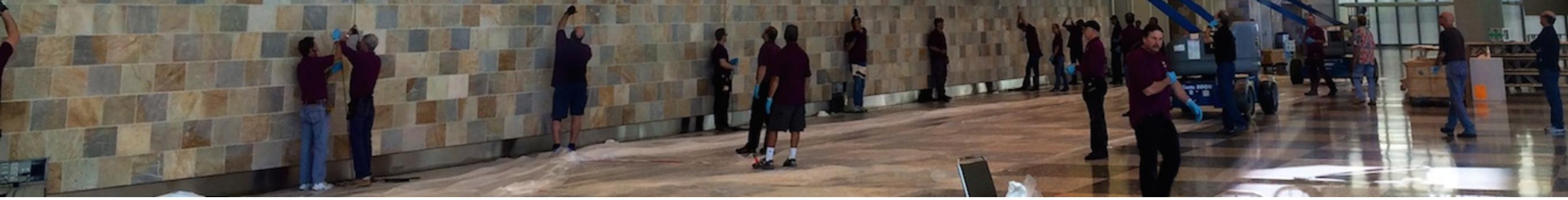
"KIT"





WWDC14

Write the code. Change the world.



# REACTIVEUI PRIMITIVES

**WhenAny:** Observe properties of objects

**ToProperty:** creates read-only derived properties from Observables

**Command:** an abstraction to invoke and marshal an asynchronous method

**Observable Lists:** mutable lists whose changes can be Observed

# WHENANY

```
this.WhenAny(x => x.Sha, x => x.Value)
    .Select(sha =>
        sha == null || sha == unknownSha ?
        "(n/a)" :
        sha.Length > 7 ?
        sha.Substring(0, 7) :
        sha)
    .ToProperty(this, x => x.ShortSha, out shortSha);
```

# COMMANDS

```
LoadUsersAndAvatars = ReactiveCommand.CreateAsyncTask(async _ => {
    var users = await LoadUsers();

    foreach(var u in users) {
        u.Avatar = await LoadAvatar(u.Id);
    }

    return users;
});

LoadUsersAndAvatars.ToProperty(this, x => x.Users, ref users);

LoadUsersAndAvatars.ThrownExceptions
    .Subscribe(ex => this.Log().WarnException("Failed to load users", ex));
```

# OBSERVABLE COLLECTIONS

```
public class TweetsListViewModel : ReactiveObject
{
    ReactiveList<Tweet> Tweets = new ReactiveList<Tweet>();

    IReactiveDerivedList<TweetTileViewModel> TweetTiles;
    IReactiveDerivedList<TweetTileViewModel> VisibleTiles;

    public TweetsListViewModel()
    {
        TweetTiles = Tweets.CreateDerivedCollection(
            x => new TweetTileViewModel() { Model = x },
            x => true,
            x => x.CreatedAt);

        VisibleTiles = TweetTiles.CreateDerivedCollection(
            x => x,
            x => !x.isHidden);
    }
}
```

# ELM AND REACTIVEUI COMPARISONS

- Both effectively use time-varied values (i.e. Behaviors)
- Elm helps you with scheduling and ordering, Rx makes you think about it explicitly
- Elm disallows signals-of-signals, Rx lets you use them (but doesn't provide any help with it)

# SOME GREAT THINGS



**State! State! State! State!**

# A CONSISTENT MODEL OF ASYNCHRONY

- Imperative code has 1030534x ways to model asynchrony (callbacks, events, promises)
- Modeling asynchronous code in tests is actually possible

# A POWERFUL MODEL OF ASYNCHRONY

- Describing complicated policies of asynchronous operations become sane to write
- Discrete UI concepts can be written succinctly

	Alex Jordan	bugfixes	17 days ago		
	Alex Jordan	awkward because i used the wrong sample program for...	1 month ago		
	Alex Jordan	i lied. more errors. also, start test program	1 month ago		
	Alex Jordan	get us to a point with no errors!	1 month ago		

▶  tools\TelemetryGenAutonomous\secondData\motor1data.txt NEW

▶  tools\TelemetryGenAutonomous\secondData\motor2data.txt NEW

▶  tools\TelemetryGenAutonomous\secondData\motor3data.txt NEW

▶  tools\TelemetryGenAutonomous\secondData\motor4data.txt NEW

◀ UNDO

Discarded changes for 'README2'

# EXPRESSING COMPLICATED UI INTENT

```
// Dismiss the undo flash after a certain amount of time.  
// NB: The logic here is, "Show the flash for at *least* 5  
// seconds. If the user does any UI action after that, *or*  
// it's been a super long time, dismiss the flash"  
Observable.Timer(TimeSpan.FromSeconds(5.0), RxApp.MainThreadScheduler)  
    .SelectMany(_ =>  
        Observable.Amb(  
            anyUIAction.Take(1),  
            Observable.Timer(TimeSpan.FromSeconds(20.0), RxApp.MainThreadScheduler).SelectUnit()))  
    .TakeUntil(ex.DoUndo)  
    .Where(x => ex.Ok.CanExecute(null))  
    .Subscribe(_ => ex.Ok.Execute(null));
```

TIME TRAVEL TESTING

[Fact]

```
public void CommitterDateTakesPrecedenceForRelativeTime()
{
    new TestScheduler().With(sched => {
        var c = new CommitModel(Mock.Of<IAvatarProvider>());
        sched.Start();

        c.AuthorDate = sched.Now;
        sched.Start();

        Assert.Equal("just now", c.FriendlyRelativeCommitTime);

        sched.AdvanceBy(TimeSpan.FromMinutes(5));
        c.CommitterDate = sched.Now;
        sched.AdvanceBy(TimeSpan.FromMinutes(5));
        c.UpdateCommitTime.Execute(null);
        sched.Start();

        Assert.Equal("5 minutes ago", c.FriendlyRelativeCommitTime);
    });
}
```

"I HAVE NO IDEA HOW  
ANYONE CAN WRITE APPS  
WITHOUT RX"

— Every developer I've met who gets good at RxUI /  
RAC



SOME NOT GREAT  
THINGS

# TEAM EDUCATION

- Most developers haven't used RxUI / RAC before, every developer we hire has to take The RAC Class™
- New developers will fall back onto writing imperative code

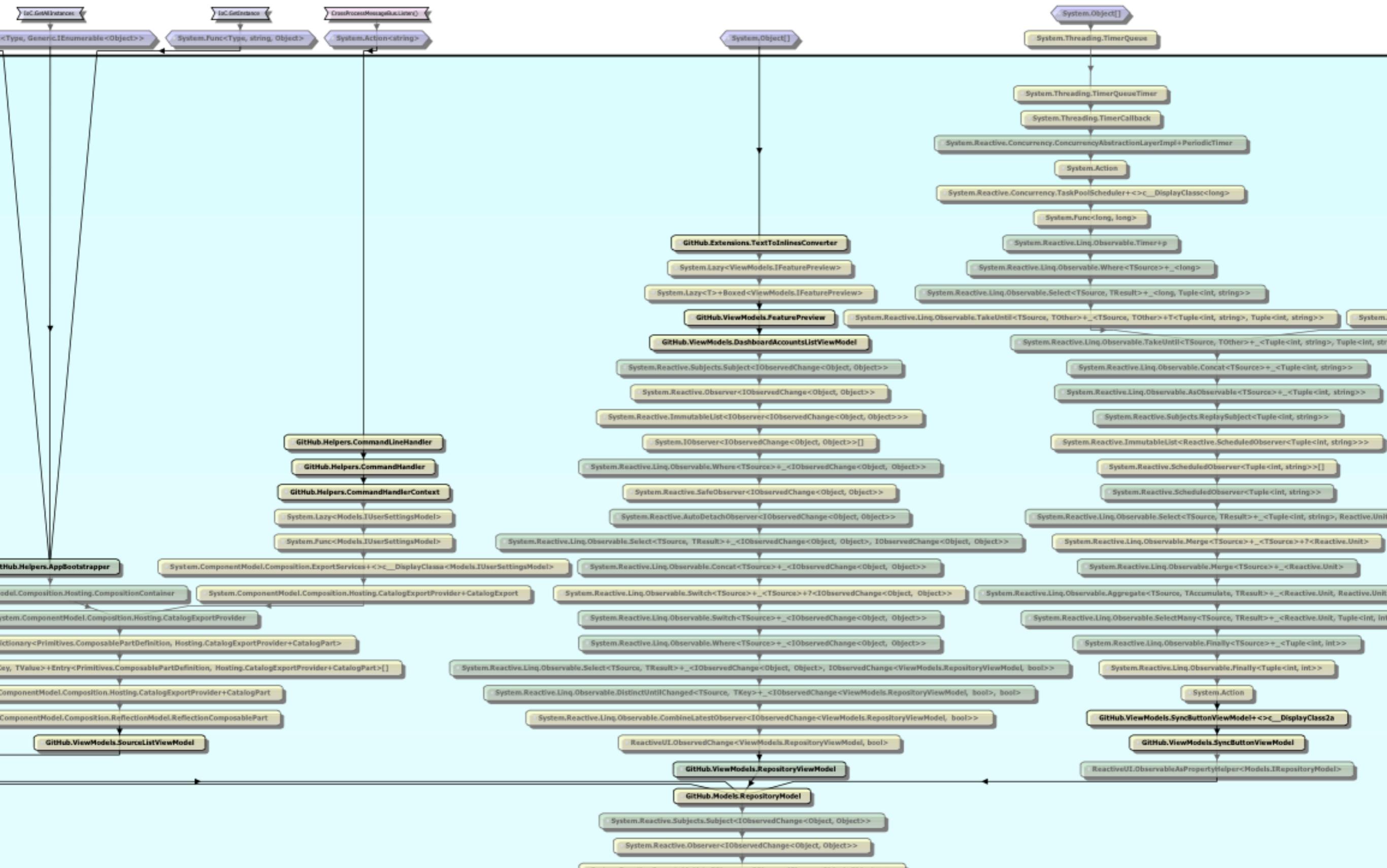
# UI FRAMEWORKS CARE ABOUT THREADS

(or really, just one thread)

# SPACE AND TIME (LEAKS)



CREATING LONG CHAINS OF  
EVENTS IS CRUISE CONTROL  
FOR MEMORY LEAKS



VIRTUALIZING LISTS ARE  
REALLY SENSITIVE TO  
ALLOCATIONS

RX LOVES ALLOCATIONS

LONG CALL STACKS ARE SCARY

```
<Error>: 0x2c5000 handle_connection: Could not receive request from host.
<Error>: HID: The 'Passive' connection 'FormsTemplateiOS' access to protected services is denied.
<Error>: Stacktrace:
<Error>: at <unknown> <0xffffffff>
<Error>: at (wrapper unknown) object.gsharedvt_out () <0xffffffff>
<Error>: at ReactiveUI.ReactiveNotifyPropertyChangedMixin.<SubscribeToExpressionChain`2>m__1<!0, !1> (ReactiveUI.IObservedChange`2<object, object>) [0x00000] in /Users/paul/code/reactiveui/ReactiveUI/ReactiveNotifyPropertyChangedMixin.cs:111
<Error>: at System.Reactive.Linq.ObservableImpl.Where`1/_<OnNext (TSource) <0x000b3>
<Error>: at System.Reactive.Linq.ObservableImpl.Switch`1/_<Iter.OnNext (TSource) <0x00153>
<Error>: at System.Reactive.Linq.ObservableImpl.Concat`1/_<OnNext (TSource) <0x000a3>
<Error>: at System.Reactive.Linq.ObservableImpl.ToObservable`1/_<LoopRec
(System.Reactive.Linq.ObservableImpl.ToObservable`1/_<State<TSource>, System.Action`1<System.Reactive.Linq.ObservableImpl.ToObservable`1/_<State<TSource>>) <0x004ef>
<Error>: at System.Reactive.Concurrency.Scheduler/<>c__DisplayClass50`1.<InvokeRecl>b__4d (TState) <0x000d3>
<Error>: at System.Reactive.Concurrency.Scheduler.InvokeRecl<TState>
(System.Reactive.Concurrency.IScheduler, System.Reactive.Concurrency.Scheduler/Pair`2<TState, System.Action`2<TState, System.Action`1<TState>>) <0x00203>
<Error>: at (wrapper unknown) object.gsharedvt_out () <0xffffffff>
<Error>: at (wrapper delegate-invoke) System.Func`3<!0, !1, !2>.invoke_TResult_T1_T2 (!0,!1) <0xffffffff>
<Error>: at (wrapper unknown) object.gsharedvt_in () <0xffffffff>
<Error>: at (wrapper unknown) object.gsharedvt_out () <0xffffffff>
<Error>: at System.Reactive.Concurrency.ImmediateScheduler.Schedule<!0> (!!0, System.Func`3<System.Reactive.Concurrency.IScheduler, !!0, System.IDisposable>) <IL 0x00015, 0x00157>
<Error>: at (wrapper unknown) object.gsharedvt_in () <0xffffffff>
<Error>: at System.Reactive.Concurrency.Scheduler.Schedule<TState> (System.Reactive.Concurrency.IScheduler, TState, System.Action`2<TState, System.Action`1<TState>>) <0x0020b>
<Error>: at System.Reactive.Linq.ObservableImpl.ToObservable`1/_<Run () <0x00403>
<Error>: at System.Reactive.Linq.ObservableImpl.ToObservable`1.Run (System.IObserver`1<TSource>, System.IDisposable, System.Action`1<System.IDisposable>) <0x000ff>
<Error>: at System.Reactive.Producer`1.SubscribeRaw (System.IObserver`1<TSource>, bool) <0x004c7>
<Error>: at System.ObservableExtensions.SubscribeSafe<T> (System.IObservable`1<T>, System.IObserver`1<T>) <0x002af>
<Error>: at System.Reactive.TailRecursiveSink`1.MoveNext () <0x00d2f>
<Error>: at System.Reactive.Concurrency.AsyncLock.Wait (System.Action) <IL 0x000a3, 0x0049b>
<Error>: at System.Reactive.TailRecursiveSink`1.<Run>b__0 (System.Action) <0x00117>
<Error>: at System.Reactive.Concurrency.Scheduler.<Schedule>b__45 (System.Action`1<System.Action>, System.Action`1<System.Action`1<System.Action>>) <IL 0x00026, 0x00167>
<Error>: at System.Reactive.Concurrency.Scheduler/<>c__DisplayClass50`1.<InvokeRecl>b__4d (TState) <0x000d3>
<Error>: at System.Reactive.Concurrency.Scheduler.InvokeRecl<TState>
(System.Reactive.Concurrency.IScheduler, System.Reactive.Concurrency.Scheduler/Pair`2<TState, System.Action`2<TState, System.Action`1<TState>>) <0x00203>
<Error>: at (wrapper unknown) object.gsharedvt_out () <0xffffffff>
<Error>: at (wrapper delegate-invoke) System.Func`3<!0, !1, !2>.invoke_TResult_T1_T2 (!0,!1) <0xffffffff>
<Error>: at (wrapper unknown) object.gsharedvt_in () <0xffffffff>
<Error>: at (wrapper unknown) object.gsharedvt_out () <0xffffffff>
<Error>: at System.Reactive.Concurrency.ImmediateScheduler.Schedule<!0> (!!0, System.Func`3<System.Reactive.Concurrency.IScheduler, !!0, System.IDisposable>) <IL 0x00015, 0x00157>
<Error>: at (wrapper unknown) object.gsharedvt_in () <0xffffffff>
<Error>: at System.Reactive.Concurrency.Scheduler.Schedule<TState> (System.Reactive.Concurrency.IScheduler, TState, System.Action`2<TState, System.Action`1<TState>>) <0x0020b>
<Error>: at System.Reactive.Concurrency.Scheduler.Schedule (System.Reactive.Concurrency.IScheduler, System.Action`1<System.Action>) <IL 0x0003b, 0x0020f>
<Error>: at System.Reactive.TailRecursiveSink`1.Run (System.Collections.Generic.IEnumerable`1<System.IObservable`1<TSource>>) <0x0036b>
<Error>: at System.Reactive.Linq.ObservableImpl.Concat`1.Run (System.IObserver`1<TSource>, System.IDisposable, System.Action`1<System.IDisposable>) <0x000fb>
<Error>: at System.Reactive.Producer`1.SubscribeRaw (System.IObserver`1<TSource>, bool) <0x004c7>
<Error>: at System.ObservableExtensions.SubscribeSafe<T> (System.IObservable`1<T>, System.IObserver`1<T>) <0x002af>
<Error>: at System.Reactive.Linq.ObservableImpl.Switch`1/_<OnNext (System.IObserver`1<TSource>) <0x002ff>
```

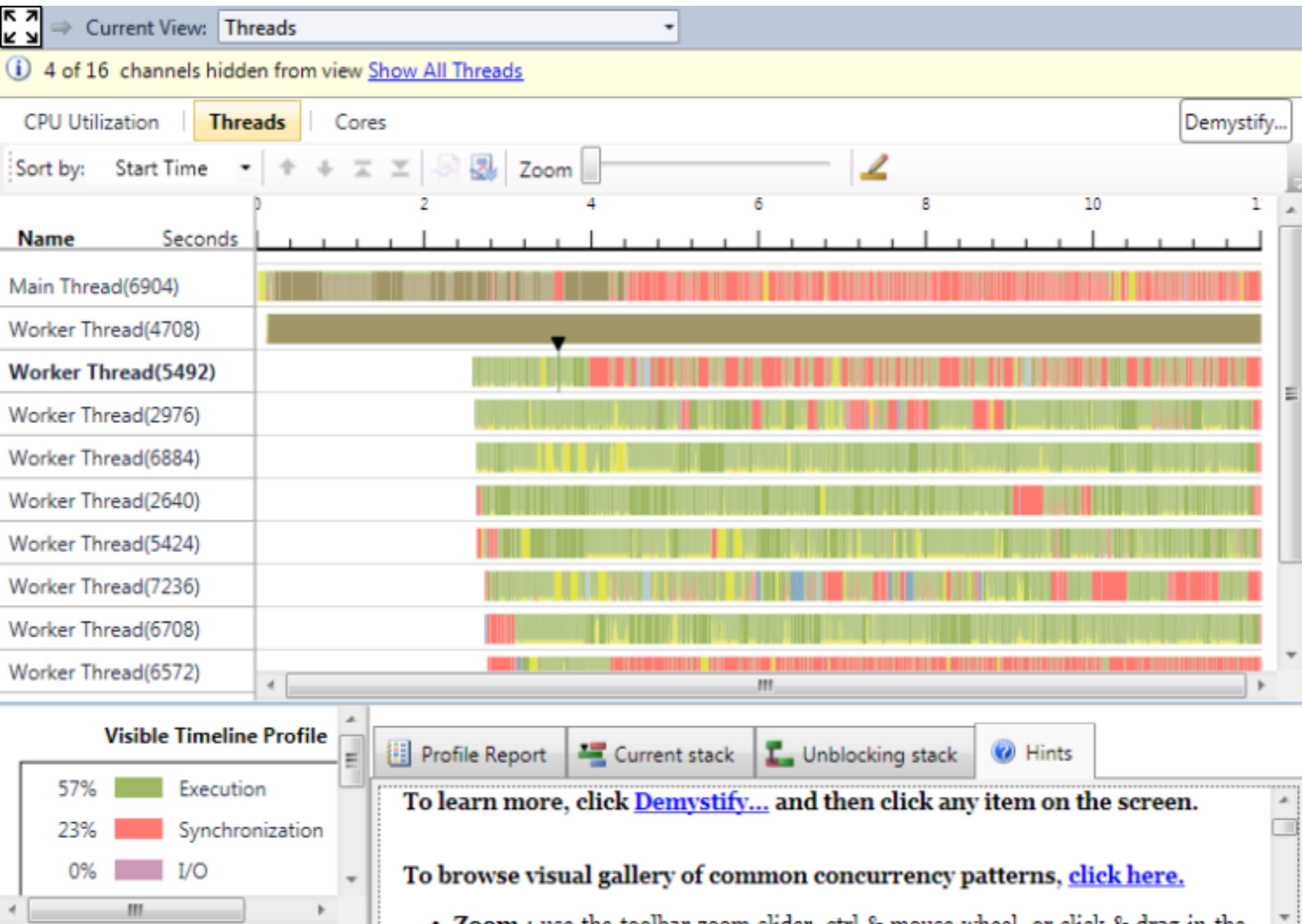
```
<Error>: at System.ObservableExtensions.SubscribeSafe<T> (System.IObservable`1<T>,System.IObserver`1<T>) <0x002af>
<Error>: at System.Reactive.Linq.ObservableImpl.Switch`1/_OnNext (System.IObservable`1<TSource>) <0x002ff>
<Error>: at System.Reactive.Linq.ObservableImpl.Select`2/_OnNext (TSource) <0x001df>
<Error>: at System.Reactive.Linq.ObservableImpl.Return`1/_Invoke () <0x000a7>
<Error>: at System.Reactive.Concurrency.Scheduler.Invoke (System.Reactive.Concurrency.IScheduler, System.Action) <IL 0x00001, 0x00073>
<Error>: at System.Reactive.Concurrency.ImmediateScheduler.Schedule<TState> (TState, System.Func`3<System.Reactive.Concurrency.IScheduler, TState, System.IDisposable>) <0x0010b>
<Error>: at System.Reactive.Concurrency.Scheduler.Schedule (System.Reactive.Concurrency.IScheduler, System.Action) <IL 0x0002a, 0x001db>
<Error>: at System.Reactive.Linq.ObservableImpl.Return`1/_Run () <0x000f7>
<Error>: at System.Reactive.Linq.ObservableImpl.Return`1.Run (System.IObserver`1<TResult>, System.IDisposable, System.Action`1<System.IDisposable>) <0x000ff>
<Error>: at System.Reactive.Producer`1.SubscribeRaw (System.IObserver`1<TSource>, bool) <0x004c7>
<Error>: at System.ObservableExtensions.SubscribeSafe<T> (System.IObservable`1<T>, System.IObserver`1<T>) <0x002af>
<Error>: at System.Reactive.Linq.ObservableImpl.Select`2.Run (System.IObserver`1<TResult>, System.IDisposable, System.Action`1<System.IDisposable>) <0x00147>
<Error>: at System.Reactive.Producer`1.SubscribeRaw (System.IObserver`1<TSource>, bool) <0x004c7>
<Error>: at System.ObservableExtensions.SubscribeSafe<T> (System.IObservable`1<T>, System.IObserver`1<T>) <0x002af>
<Error>: at System.Reactive.Linq.ObservableImpl.Switch`1/_Run () <0x001df>
<Error>: at System.Reactive.Linq.ObservableImpl.Switch`1.Run (System.IObserver`1<TSource>, System.IDisposable, System.Action`1<System.IDisposable>) <0x000ff>
<Error>: at System.Reactive.Producer`1.SubscribeRaw (System.IObserver`1<TSource>, bool) <0x004c7>
<Error>: at System.ObservableExtensions.SubscribeSafe<T> (System.IObservable`1<T>, System.IObserver`1<T>) <0x002af>
<Error>: at System.Reactive.Linq.ObservableImpl.Where`1.Run (System.IObserver`1<TSource>, System.IDisposable, System.Action`1<System.IDisposable>) <0x00147>
<Error>: at System.Reactive.Producer`1.SubscribeRaw (System.IObserver`1<TSource>, bool) <0x004c7>
<Error>: at System.ObservableExtensions.SubscribeSafe<T> (System.IObservable`1<T>, System.IObserver`1<T>) <0x002af>
<Error>: at System.Reactive.Linq.ObservableImpl.Select`2.Run (System.IObserver`1<TResult>, System.IDisposable, System.Action`1<System.IDisposable>) <0x00147>
<Error>: at System.Reactive.Producer`1.SubscribeRaw (System.IObserver`1<TSource>, bool) <0x004c7>
<Error>: at System.ObservableExtensions.SubscribeSafe<T> (System.IObservable`1<T>, System.IObserver`1<T>) <0x002af>
<Error>: at (wrapper unknown) object.gsharedvt_out () <0xffffffff>
<Error>: at System.Reactive.Linq.ObservableImpl.DistinctUntilChanged`2<!0, !1>.Run (System.IObserver`1<!0>, System.IDisposable, System.Action`1<System.IDisposable>) <IL 0x00017, 0x00193>
<Error>: at System.Reactive.Producer`1.SubscribeRaw (System.IObserver`1<TSource>, bool) <0x004c7>
<Error>: at System.ObservableExtensions.SubscribeSafe<T> (System.IObservable`1<T>, System.IObserver`1<T>) <0x002af>
<Error>: at (wrapper unknown) object.gsharedvt_out () <0xffffffff>
<Error>: at System.Reactive.Linq.ObservableImpl.Select`2<!0, !1>.Run (System.IObserver`1<!1>, System.IDisposable, System.Action`1<System.IDisposable>) <IL 0x0001f, 0x001c3>
<Error>: at System.Reactive.Producer`1<bool>.Run (System.Reactive.Concurrency.IScheduler, System.Reactive.Producer`1/State<bool>) <IL 0x00027, 0x0018b>
<Error>: at (wrapper unknown) object.gsharedvt_out () <0xffffffff>
<Error>: at System.Reactive.Concurrency.ScheduledItem`2<!0, !1>.InvokeCore () <IL 0x00012, 0x000fb>
<Error>: at System.Reactive.Concurrency.ScheduledItem`1<System.TimeSpan>.Invoke () <IL 0x00014, 0x000c3>
<Error>: at System.Reactive.Concurrency.CurrentThreadScheduler/Trampoline.Run (System.Reactive.Concurrency.SchedulerQueue`1<System.TimeSpan>) <IL 0x00041, 0x002db>
<Error>: at System.Reactive.Concurrency.CurrentThreadScheduler.Schedule<!10> (!!0, System.TimeSpan, System.Func`3<System.Reactive.Concurrency.IScheduler, !!0, System.IDisposable>) <IL 0x00047, 0x0039b>
<Error>: at (wrapper unknown) object.gsharedvt_in () <0xffffffff>
<Error>: at (wrapper unknown) object.gsharedvt_out () <0xffffffff>
<Error>: at System.Reactive.Concurrency.LocalScheduler.Schedule<!10> (!!0, System.Func`3<System.Reactive.Concurrency.IScheduler, !!0, System.IDisposable>) <IL 0x00016, 0x0019f>
<Error>: at (wrapper unknown) object.gsharedvt_in () <0xffffffff>
<Error>: at System.Reactive.Producer`1<bool>.SubscribeRaw (System.IObserver`1<bool>, bool) <IL 0x0007b, 0x003af>
<Error>: at System.Reactive.Producer`1<bool>.Subscribe (System.IObserver`1<bool>) <IL 0x00011, 0x000d7>
<Error>: at System.ObservableExtensions.Subscribe<bool> (System.IObservable`1<bool>, System.Action`1<bool>) <IL 0x0002d, 0x001bb>
<Error>: at ReactiveTest.App.Get MainPage () [0x00007] in d:\Dev\ReactiveTest\ReactiveTest\ReactiveTest\App.cs:17
<Error>: at ReactiveTest.iOSAppDelegate.FinishedLaunching (MonoTouch.UIKit.UIApplication, MonoTouch.Foundation.NSDictionary) [0x0001c] in d:\Dev\ReactiveTest\ReactiveTest\ReactiveTest.iOS\AppDelegate.cs:34
<Error>: at /var/mobile/Device/Cache/mono/mcs/class/System/IObservable.cs:16
```

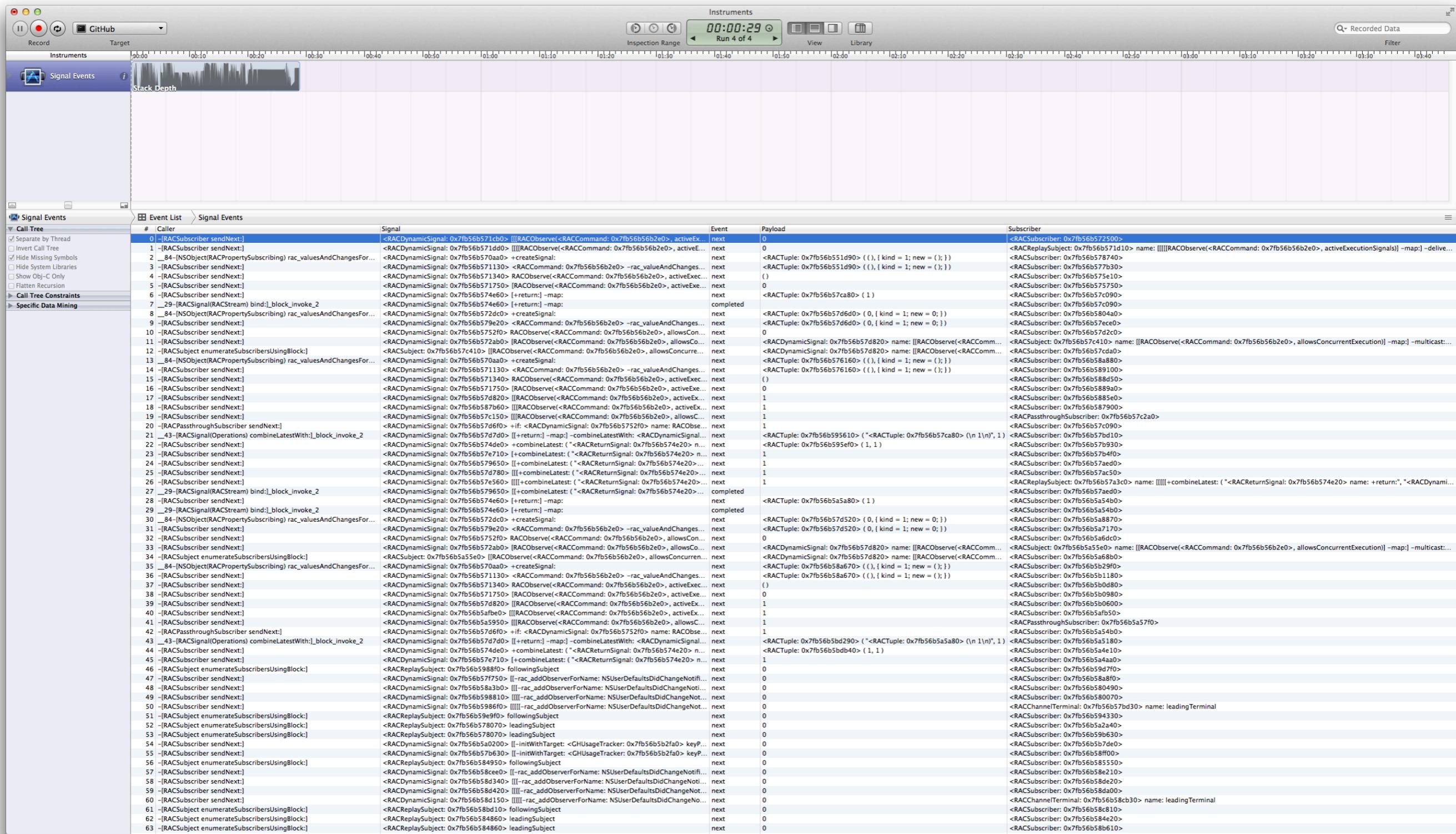
```
<Error>: at System.Reactive.Producer`1<bool>.SubscribeRaw (System.IObserver`1<bool>,bool) <IL 0x0007b, 0x003af>
<Error>: at System.Reactive.Producer`1<bool>.Subscribe (System.IObserver`1<bool>) <IL 0x00011, 0x000d7>
<Error>: at System.ObservableExtensions.Subscribe<bool> (System.IObservable`1<bool>,System.Action`1<bool>) <IL 0x0002d, 0x001bb>
<Error>: at ReactiveTest.App.Get MainPage () [0x00007] in d:\Dev\ReactiveTest\ReactiveTest\ReactiveTest\App.cs:17
<Error>: at ReactiveTest.iOS.AppDelegate.FinishedLaunching (MonoTouch.UIKit.UIApplication,MonoTouch.Foundation.NSDictionary) [0x0001c] in
d:\Dev\ReactiveTest\ReactiveTest\ReactiveTest.iOS\AppDelegate.cs:34
<Error>: at (wrapper runtime-invoke) object.runtime_invoke_dynamic (intptr,intptr,intptr,intptr) <0xffffffff>
<Error>: at <unknown> <0xffffffff>
<Error>: at (wrapper managed-to-native) MonoTouch.UIKit.UIApplication.UIApplicationMain (int,string[],intptr,intptr) <0xffffffff>
<Error>: at MonoTouch.UIKit.UIApplication.Main (string[],string,string) [0x0004c] in /Developer/MonoTouch/Source/monotouch/src/UIKit/UIApplication.cs:39
<Error>: at ReactiveTest.iOS.Application.Main (string[]) [0x00001] in d:\Dev\ReactiveTest\ReactiveTest\ReactiveTest.iOS\Main.cs:17
<Error>: at (wrapper runtime-invoke) object.runtime_invoke_dynamic (intptr,intptr,intptr,intptr) <0xffffffff>
```

TOOLING CAN HELP US  
OUT

# LOGGING AND TRACING

- Setting up a good logging framework is critical
  - Log Signals that end in an error
- Signals are natively dual-point logging (i.e. they represent a Span)





RxSpySession.cs ObservableDetails.xaml.cs RxObservab...ewModel.cs ObservableDetails.xaml OnError

**RxSpy** file:///C:/Users/Markus/Documents/GitHub/RxSpy/RxSpy.TestConsole/bin/Debug/

**RxSpy**

Tracked observables

#	Name	Status	Parents	Children	Ancestors	Descendants	Values	Tag	Created in
5	Throw<string>	Error	0	1	0	4	0		Program.Main (string[] args) in Program.cs:33
6	Return<string>	Completed	0	1	0	4	1		Program.Main (string[] args) in Program.cs:33
7	Catch<string>	Completed	2	1	2	3	1		Program.Main (string[] args) in Program.cs:33
8	CombineLatest<string>	Active	3	1	8	2	71		Program.Main (string[] args) in Program.cs:37
9	Select<IList<string>, string>	Active	1	1	9	1	71		Program.Main (string[] args) in Program.cs:38
10	AnonymousObserver<string>	Active	1	0	10	0	0		

Details

Signals	Parents	Children	Errors
Received	Thread	Type	Value
00:00:01.0620000	0	string	"---, Bar, "
00:00:02.0620000	0	string	"---, Bar, "
00:00:02.0620000	0	string	"---, Baz, "
00:00:03.0620000	0	string	"---, Baz, "
00:00:04.0620000	0	string	"---, Baz, "

297 signals, 1 errors. 5 signals per second.

```

var obs2 = obs1.Select(x =>
    var obs3 = obs1.Select(x =>
        var obs4 = obs2.Where(x =>
            var obsErr = Observable.Throw<string>(x);
        );
        return obs4;
    );
    return obs3;
);
var toJoin = new List<IObservable<string>>{ obs3, obs4, obsErr };

```

**AnonymousObserver<string>**

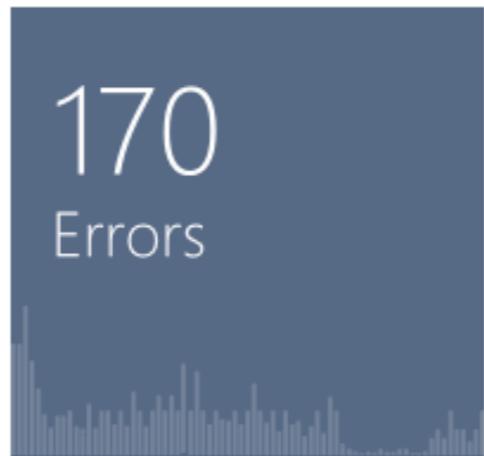
```

graph TD
    Timer[Timer 44] --> Select1[Select<long, string> 44]
    Timer --> Throw[Throw<string> 0]
    Timer --> Return[Return<string> 1]
    Select1 --> Where[Where<string> 29]
    Throw --> Catch[Catch<string> 1]
    Catch --> CombineLatest[CombineLatest<string> 71]
    CombineLatest --> Select2[Select<IList<string>, string> 71]
    Select2 --> Observer[AnonymousObserver<string> 0]

```

## MainWindow

## Overview ▾



## Activity



# THANKS!

— @paulcbetts