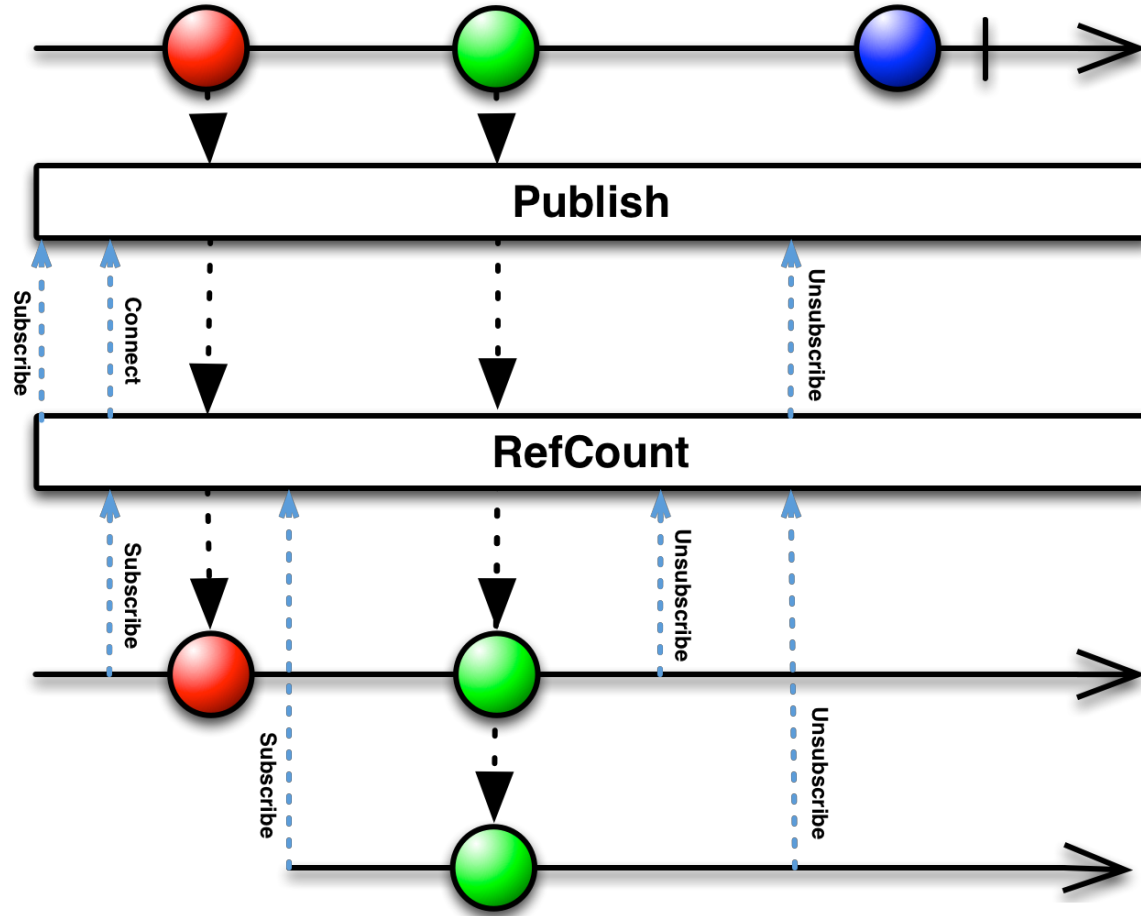


30.RefCount

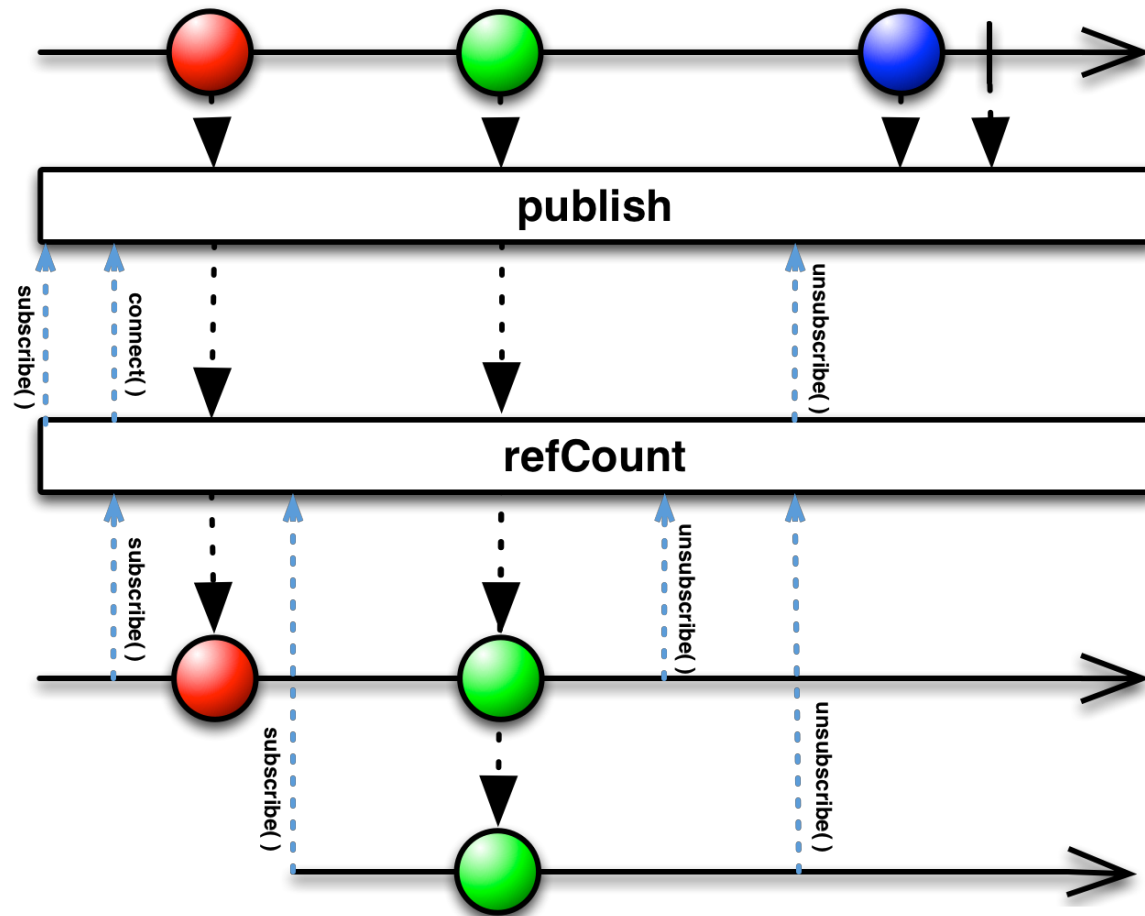
RefCount 示意图

让一个可连接的 Observable 行为像普通的Observable

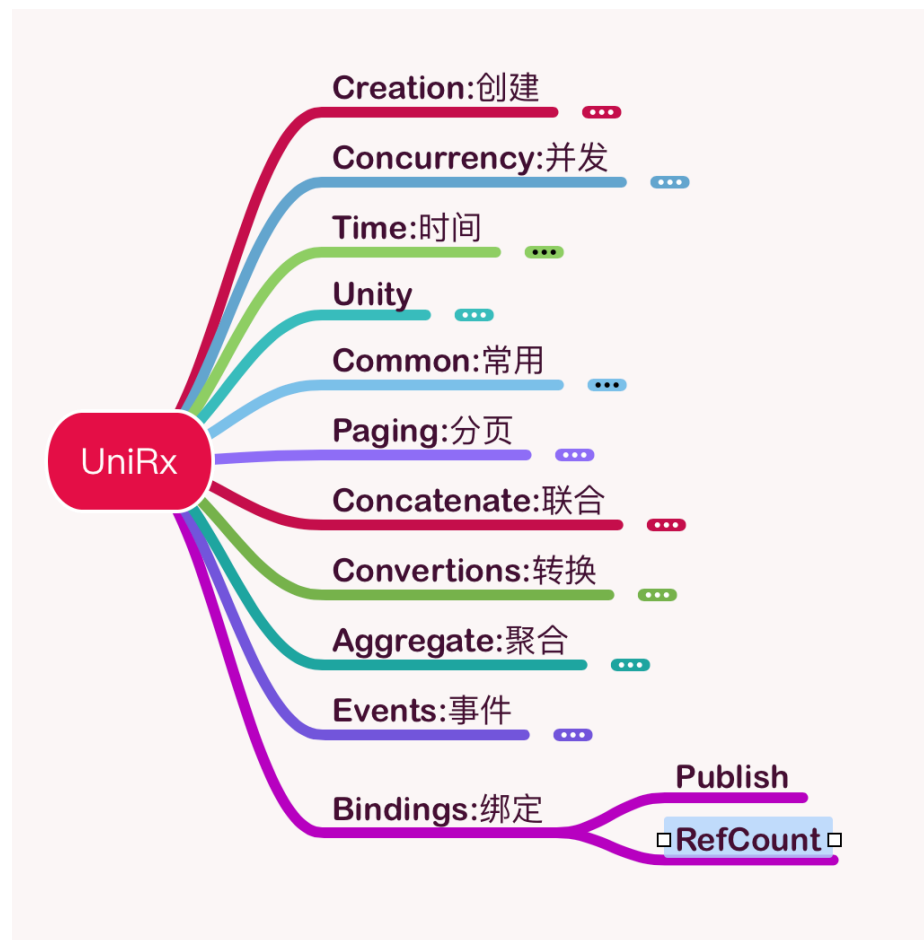


可连接的 Observable (*connectable Observable*) 与普通的 Observable 差不多，不过它并不会在被订阅时开始发射数据，而是直到使用了Connect操作符时才会开始。用这种方法，你可以在任何时候让一个 Observable 开始发射数据。

RefCount操作符把从一个可连接的 Observable 连接和断开的过程自动化了。它操作一个可连接的 Observable，返回一个普通的 Observable。当第一个订阅者订阅这个Observable时，RefCount 连接到下层的可连接 Observable。RefCount 跟踪有多少个观察者订阅它，直到最后一个观察者完成才断开与下层可连接Observable的连接。



RefCount 所在知识地图中的位置



RefCount 示例代码

```
/******  
 * http://sikiedu.com liangxie  
******/  
  
using System;  
using UniRx;  
using UnityEngine;  
  
namespace UniRxLesson  
{  
    public class RefCountExample : MonoBehaviour  
    {  
        void Start()  
        {  
            var period = TimeSpan.FromSeconds(1);  

```

```

var observable = Observable.Interval(period)
                                .Do(1 => Debug.LogFormat("Publishing
{0}", 1)) //side effect to show it is running
                                .Publish()
                                .RefCount();

Observable.Timer(TimeSpan.FromSeconds(5.0f)).Subscribe(_ =>
{
    Debug.Log("@@@ subscribe @@@");
    //observable.Connect(); Use RefCount instead now
    var subscription = observable.Subscribe(i =>
Debug.LogFormat("subscription : {0}", i));

    Observable.Timer(TimeSpan.FromSeconds(5.0f)).Subscribe(__ =>
    {
        Debug.Log("@@@ unsubscribe @@@");
    }
});

```



```

        subscription.Dispose();

        observable.Subscribe(i => Debug.LogFormat("subscription 2: {0}",
i));

    });

}

}

}

```

输出结果为:

```

// 5 秒后
@@@ subscribe @@@
Publishing 0
subscription : 0
Publishing 1
subscription : 1

```

```
Publishing 2  
subscription : 2  
Publishing 3  
subscription : 3  
@@@ unsubscribe @@@  
Publishing 0  
subscription 2: 0  
Publishing 1  
subscription 2: 1  
Publishing 2  
subscription 2: 2  
Publishing 3  
subscription 2: 3  
...
```

除此之外，还有一个 Share 操作符：

```
/// <summary>  
/// same as Publish().RefCount()
```

```
/// </summary>  
public static IObservable<T> Share<T>(this IObservable<T> source)  
{  
    return source.Publish().RefCount();  
}
```