

# Cancellation

Ian Griffiths

<http://www.interact-sw.co.uk/iangblog/>

ian@interact-sw.co.uk



# Cancellation Model

- Consistency
- Chainability
- Blocking and asynchronous work
- Thread-based and threadless work
- Cooperative



# Cooperative Cancellation

- Optional
- Asynchronous
- Might not happen



# CancellationToken

- **Argument for cancellable operations**
  - Single-use
- **Enables recipients to discover cancellation request**
  - Polling
  - Notification
  - WaitHandle
- **Does not allow initiation**
  - Use CancellationTokenSource



```
using (var w = new WebClient())
{
    var dtcs = new TaskCompletionSource<string>();
    w.DownloadStringCompleted += (s, e) =>
    {
        if (e.Error != null) { dtcs.SetException(e.Error); }
        else if (e.Cancelled) { dtcs.SetCanceled(); }
        else { dtcs.SetResult(e.Result); }
    };

    string result;
    using (ct.Register(() => w.CancelAsync()))
    {
        result = w.DownloadStringAsync("http://example.com/foo");
        dtcs.Task.Wait();
    }

    ct.ThrowIfCancellationRequested();
    string nextUrl = DoDbLookup(result);

    ct.ThrowIfCancellationRequested();
    dtcs = new TaskCompletionSource<string>();
    using (ct.Register(() => w.CancelAsync()))
    {
        result = w.DownloadStringAsync(nextUrl);
        dtcs.Task.Wait();
    }
}
```

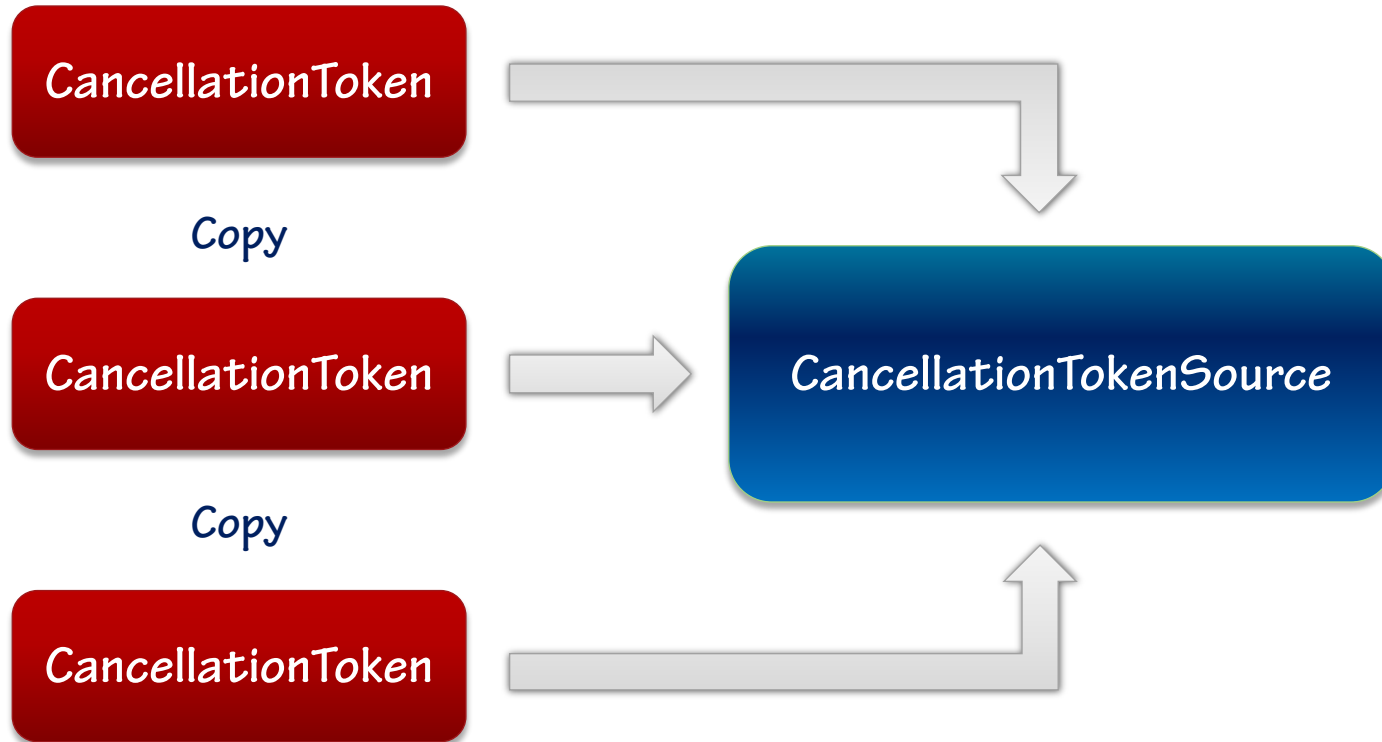


# SynchronizationContext

- `CancellationToken.Register(callback, state, true);`



# CancellationToken is a Value Type



# Tasks and Cancellation

- **Thread-based**
  - Can prevent task from starting
  - Special handling of `OperationCanceledException`
  - `Wait` and `Result` throw `TaskCanceledException`
- **`Task.WhenAll`**
- **Continuations not canceled**
  - Pass token to `ContinueWith` if you want that





# Cancellation and Deadlock

- **Rules for callbacks:**
  - Run quickly
  - Be wary of `CancellationTokenRegistration.Dispose`
  - Don't use `SynchronizationContext`



# Linked Tokens

- **CancellationTokenSource.CreateLinkedTokenSource**
  - Takes any number of CancellationToken
  - Returns CancellationTokenSource



# Summary

- Goals
- Token vs token source
- Polling, notification, and WaitHandle
- Tasks

