

I'll Get Back to You

Task, Await and Asynchronous Programming

Jeremy Clark
Developer Betterer
jeremybytes.com

Level: Introductory / Intermediate



Session Survey

- Your feedback is very important to us
- Please take a moment to complete the session survey found in the mobile app
- Use the QR code or search for “Converge360 Events” in your app store
- Find this session on the Agenda tab
- Click “Session Evaluation”
- Thank you!



I'll Get Back to You

Task, Await and Asynchronous Programming

Jeremy Clark
Developer Betterer
jeremybytes.com

Level: Introductory / Intermediate



Async Revolves Around Task

Use Task directly

- Extremely powerful
- Lots of options
- Very complex

await Task

- Looks like other code
- Limited functionality
- Fits the “95% scenario”

Task<T>

- Method Returns a Task
 - Task<T> GetDataAsync()
 - “T” = result type for the result
- Task
 - Represents a concurrent operation
 - May or may not operate on a separate thread
 - Can be chained and combined

async & await

- Syntactic Wrapper Around Task
 - “await” pauses the current method until Task is complete.
 - Looks like a blocking operation
 - Does not block current thread
- “async” modifier
 - Tells the compiler to treat “await” as noted above
 - The compiled method is turned into a state machine

Task Properties (.NET 8 / 9)

- Task Properties

- IsFaulted
- IsCanceled
- IsCompleted*
- IsCompletedSuccessfully

IsCompletedSuccessfully

- .NET 8 / 9
- .NET Standard 2.1
- NOT .NET Standard 2.0
- NOT .NET Framework

**Note: Means “no longer running”
not “completed successfully”*

Task Properties (.NET Framework)

- Task Properties

- IsFaulted
- IsCanceled
- IsCompleted*
- **Status**

**Note: Means “no longer running” not “completed successfully”*

- TaskStatus

- **Canceled**
- Created
- **Faulted**
- **RanToCompletion**
- Running
- WaitingForActivation
- WaitingForChildrenToComplete
- WaitingToRun

Exception Handling

- `AggregateException`
 - Tree structure
- `Flatten()`
 - Flattens the tree structure to a single level

Cancellation

- CancellationToken is ReadOnly
 - new CancellationToken(true)
 - new CancellationToken(false)
- CancellationTokenSource
 - var cts = new CancellationTokenSource()
 - var token = cts.Token
 - cts.Cancel()

VSLive! 2-Day Hands-On Training Seminar

Asynchronous and Parallel Programming in C#

June 24-25, 2025 | Virtual Training

REGISTER TODAY

In-Depth Training **for Developers**

Session Survey

- Your feedback is very important to us
- Please take a moment to complete the session survey found in the mobile app
- Use the QR code or search for “Converge360 Events” in your app store
- Find this session on the Agenda tab
- Click “Session Evaluation”
- Thank you!





Thank You!

Jeremy Clark

- jeremybytes.com
- jeremy@jeremybytes.com
- [@jeremybytes](https://twitter.com/jeremybytes)

<https://github.com/jeremybytes/vslive2025-lasvegas>