**C# Job System Overview**

**How the C# Job System works**

The Unity C# Job System allows users to write [multithreaded code](https://en.wikipedia.org/wiki/Multithreading_(computer_architecture)) that interacts well with the rest of Unity and makes it easier to write correct code.

Writing multithreaded code can provide high-performance benefits. These include significant gains in frame rate. Using the Burst compiler with C# jobs gives you improved [code generation](https://en.wikipedia.org/wiki/Code_generation_(compiler)) quality, which also results in substantial reduction of battery consumption on mobile devices.

An essential aspect of the C# Job System is that it integrates with what Unity uses internally (Unity’s native job system). User-written code and Unity share [worker threads](https://docs.microsoft.com/en-us/cpp/parallel/multithreading-creating-worker-threads). This cooperation avoids creating more threads than [CPU cores](https://en.wikipedia.org/wiki/Multi-core_processor), which would cause contention for CPU resources.

For more information, watch the talk [Unity at GDC - Job System & Entity Component System](https://www.youtube.com/watch?v=kwnb9Clh2Is&t=1s).