
Subject: Re: Coroutines package for U++
Posted by [Oblivion](#) on Mon, 12 Dec 2022 16:42:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello zsolc

Quote: Is it possible some easy way to write a function / method to start rendering some big images or PDFs on other threads and awaiting them, to complete, not blocking the main (GUI) thread, that started it?

In my experience the real question would be: Will it worth it?

My answer would be: No. Because what you describe is already achievable by using `Upp::CoWork` or `Upp::AsyncWork`. You can offload a rendering operation using these tools without blocking the gui, check their status and wait for them to join in non-blocking way and cancel them when you need to (`AsyncWork` has a `Cancel` method). In fact, I have a small utility (app) called `SshGet` (an MT Sftp/Scp transfer queue), where I do something very similar to what you describe (instead of rendering stuff, I am transferring them).

However, the direct answer to your questions is that it is partially possible. `cpp/coroutines` have a concept called `awaitables`. You can write an `awaitable` type (class or functor) and pass the coroutine handle among threads in a lock free way via `awaitable`. But 1) It is complicated, 2) It won't solve the real problem: termination of a thread. Plus, what you achieve in the end will seem to be what `CoWork` et al. can already do.

This is the reason why I opted to implement a light-weight coroutines interface (only routine and generator types and not the `awaitable` type.) This makes things simple and clean.

Best regards,
Oblivion