
Subject: Re: RE: Job package: A scope-bound worker thread for non-blocking operations.

Posted by [Oblivion](#) on Tue, 19 Sep 2017 10:31:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

How about adding An Async/Result pair, the way you suggested.

ASync will be an improved version of the example you provided with (using CoWork), addition of result id generator (for cancellation), exception handling:

Result will be a wrapper for future where we can do such things as follows:

```
auto result1 = Async([=]{ return GetDivisors(); })
while(!result.IsReady())
    ; // Do something;
Cout() << result.IsValid() ? result.Get() : result.GetErrorDesc();

// and of course...
Cout() << Async([=]{ return "Hello World\n"; }).Get();
```

Result's Interface may be consisted of:

Result::Get();

Result::IsReady();

Result::IsValid();

Result::GetError();

Result::GetErrorDesc();

Result::GetId();

Result::Cancel();

If you'd like, I can implement it this way. (IMO Result is a more meaningful name, but naming can change...)

Best regards,
Oblivion
