

Cancelling a Windows Runtime asynchronous operation, part 5: C++/WinRT

devblogs.microsoft.com/oldnewthing/20200707-00

July 7, 2020



Raymond Chen

Last time, we looked at [how task cancellation is projected in C++/CX with PPL and co_await with IAsyncAction^ and IAsyncOperation^ objects](#). Our next investigation is C++/WinRT.

```
FileOpenPicker openPicker;
openPicker.FileTypeFilter().Append(L".txt");
auto pickerOp = openPicker.PickSingleFileAsync();
([](auto op) -> fire_and_forget { co_await resume_after(3s); op.Cancel(); })
(pickerOp);

StorageFile file{ nullptr };
try
{
    file = co_await pickerOp;
}
catch (hresult_canceled const&)
{
    file = nullptr;
}
catch (hresult_illegal_method_call const&)
{
    file = nullptr;
}

if (file != nullptr)
{
    DoSomething(file);
}
```

Canceling the operation after a delay is slightly tricky because we need to pass the `pickerOp` as a parameter to a captureless lambda, rather than capturing it into the lambda. We discussed the reason for this [some time ago](#).

The exception that comes out of awaiting for a canceled task is sometimes an `hresult_canceled`, and sometimes an `hresult_illegal_method_call`. The reason is that C++/WinRT defers to the `IAsyncAction` / `IAsyncOperation` to decide what exception to

raise.¹

In other words, C++/WinRT just takes the ABI result and propagates it. It doesn't try to impose its will upon the result.

You can see this in the `await_resume` for asynchronous operations:

```
template <typename Async>
struct await_adapter
{
    ...

    auto await_resume() const
    {
        return async.GetResults();
    }
};
```

Whatever exception comes out of `GetResults()` is the exception that comes out of the `co_await`.

Next time, we'll look at what happens if the `IAsyncAction` or `IAsyncOperation` was implemented by the C++/WinRT library itself.

Bonus chatter: This entire article is already obsolete. We'll learn more about it when this series wraps up.

¹ Therefore, in principle, it could be *utterly anything* since it's up to the ABI to generate the `HRESULT` that turns into an exception at the projection.

[Raymond Chen](#)

Follow

