

Cancelling a Windows Runtime asynchronous operation, part 8: C++/WinRT, revised

devblogs.microsoft.com/oldnewthing/20200710-00

July 10, 2020



Raymond Chen

Some time ago, we observed that C++/WinRT relies on the ABI result from the `GetResults()` method to report cancellation. This is different from how task cancellation is projected in C# and C++/CX. Furthermore, it's a leaky abstraction: The awaiting code needs to know how the underlying operation is implemented in order to know what exception will be raised upon cancellation.

Fortunately, that has been fixed in PR 643 (released as version 2.0.200601.2) so that C++/WinRT aligns with the other projections in how it handles cancellation: It now detects cancellation by checking the status of the operation.

Conceptually, it's a one-line fix:

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

    auto await_resume() const
    {
        if (async.Status() == AsyncStatus::Canceled) throw hresult_canceled();
        return async.GetResults();
    }
};
```

However, this adds the virtual method call `async.Status()` to a hot code path. Even worse, if this is a remote operation, the virtual method call has to cross a process boundary, which is even more expensive. (If you enable Async-Async, then this becomes a local query, but it's still virtual.)

The fix is to cache the status reported by the `Completed` callback:

```

inline void check_status_canceled(AsyncStatus status)
{
    if (status == AsyncStatus::Canceled) throw hresult_canceled();
}

template <typename Async>
struct await_adapter
{
    AsyncStatus status = AsyncStatus::Started;
    ...

    void await_suspend(std::experimental::coroutine_handle<> handle)
    {
        async.Completed([this, handler = disconnect_aware_handler{ handle }]
            (auto&&, auto&& operation_status)
            {
                status = operation_status;
                handler();
            });
    }

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

```

The code to convert a `Canceled` status into an exception was factored out, because similar fixes need to be made to other methods that react to the result of an operation.

[Raymond Chen](#)

Follow

