Are Windows Runtime asynchronous operations guaranteed to complete?

devblogs.microsoft.com/oldnewthing/20200327-00

March 27, 2020

Raymond Chen

The Windows Runtime uses asynchronous operations, which are operations which start and return immediately, and then notify you when the operation has completed. This lets you do other things while waiting for the operation. Most programming languages nowadays have built-in support for this style of programming, usually by using some variation of the keyword `await`.

Is there any guarantee that a Windows Runtime operation will eventually complete?

Is there any guarantee that *any* operation will eventually complete?

Not really.

For example, you might display a dialog box to the user by calling `MessageBox.ShowAsync`. This completes when the user responds to the dialog. But what if the user isn't there? The dialog box remains on screen indefinitely. Now, it's possible that the user might return someday, so you might argue that the operation hasn't definitely gotten stuck, because the user can always unstick it by responding to the dialog box.

The `AnimatedVisualPlayer.PlayAsync` method completes when the animation stops. This happens naturally if you ask the animation to play to the end and stop, but if you ask for a looping animation, then it doesn't stop until you manually call `Stop` to stop it. Does this mean that there's no guarantee that the `PlayAsync` will ever complete? I mean, your program can always unstick it by calling `Stop`.

Each asynchronous operation defines the conditions under which it will complete. If those conditions are never met, then it will never complete. There's nothing special about asynchronous operations here. This can happen with synchronous functions, too. If you ask `WaitForSingleObject` to wait for a handle that will never be signaled, then it will never return.

Bonus chatter: You can easily create your own Windows Runtime asynchronous operation that never completes.

```
winrt::IAsyncAction HangAsync()
{
    co_await std::experimental::suspend_always{};
}
```

The `suspend_always` object suspends and never wakes up. Awaiting it will never complete. And that means that the `IAsyncAction` you created from it will never complete.

Raymond Chen

Follow