Since clean-up functions can't fail, you have to soldier on

devblogs.microsoft.com/oldnewthing/20140807-00

August 7, 2014

Raymond Chen

Clean-up functions can't fail, so what do you do if you encounter a failure in your clean-up function?

You have to keep cleaning up.

Some people like to follow this pattern for error checking:

```
HRESULT Function()
{
  hr = SomeFunction();
  if (FAILED(hr))) goto Exit;
  hr = AnotherFunction();
  if (FAILED(hr))) goto Exit;
  ... and so on until ...
  hr = S_OK;
Exit:
    return hr;
}
```

And some like to put it inside a cute flow control macro like

```
#define CHECK_HRESULT(hr) if (FAILED(hr)) goto Exit;
```

or even

```
#define CHECK_HRESULT(f) if (FAILED(hr = (f))) goto Exit;
```

Whatever floats your boat.

But you have to be careful if using this pattern in a clean-up function, because you might end up not actually cleaning up. For example:

```
HRESULT Widget::Close()
{
    HRESULT hr;
    CHECK_HRESULT(DisconnectDoodad(m_hDoodad));
    m_hDoodad = nullptr;
    for (int i = 0; i < ARRRAYSIZE(GadgetArray); i++) {
        CHECK_HRESULT(DestroyGadget(m_rghGadget[i]));
        m_rghGadget[i] = nullptr;
    }
    hr = S_OK;
Exit:
    return hr;
}
```

What if there is an error disconnecting the doodad? (Maybe you got `RPC_E_SERVER_DIED` because the doodad lives on a remote server which crashed.) The cleanup code treats this as an error and skips destroying the gadget. But what can the caller do about this? Nothing, that's what. Eventually you get a bug that says, "On an unreliable network, we leak gadgets like crazy."

Or worse, what if you're doing this in your destructor. You have nowhere to report the error. The caller simply expects that when the object is destroyed, all its resources are released.

So release as much as you can. If something goes wrong with one of them, keep going, because there's still other stuff to clean up.

Related: Never throw an exception from a destructor.

Bonus chatter: Yes, I know that you can avoid this problem by wrapping the Doodad and Gadget handles inside a class which disconnects/destroys on destruction. That's not my point.

Raymond Chen

Follow