How can I convert a WIC bitmap to a Windows Runtime SoftwareBitmap? part 1: Via an encoded stream

devblogs.microsoft.com/oldnewthing/20230411-00

April 11, 2023

Raymond Chen

Say you have produced a bitmap with the Windows Imaging Component, also known as WIC, and you want to convert it to a Windows Runtime `SoftwareBitmap`, say, because you want to use it as a XAML `SoftwareBitmapSource` so you can display it in a XAML `BitmapImage`.

We'll be trying to solve this problem over the next few days, coming up with simpler and simpler solutions, until we get down to a one-liner. But let's not get ahead of ourselves.

Well, let's see. There are only a few Windows Runtime methods that produce a `SoftwareBitmap`. There are some conversion methods on `SoftwareBitmap` itself, but they assume you already have a `SoftwareBitmap`. Fortunately, there's also `BitmapDecoder`.

So you might come up with this:

```
namespace winrt
{
    using namespace winrt::Windows::Foundation;
    using namespace winrt::Windows::Graphics::Imaging;
    using namespace winrt::Windows::Storage::Streams;
}

winrt::IAsyncOperation<winrt::SoftwareBitmap>
    ToSoftwareBitmap(IWICBitmapSource* wicBitmap)
{
    // Boilerplate code to save the bitmap as a PNG stream
    winrt::com_ptr<::IStream> stream;
    winrt::check_hresult(
        CreateStreamOnHGlobal(nullptr, TRUE, stream.put()));
    auto bitmapEncoder = winrt::create_instance<
        IWICBitmapEncoder>(CLSID_WICPngEncoder);
    winrt::check_hresult(bitmapEncoder->Initialize(
        stream.get(), WICBitmapEncoderNoCache));
    winrt::com_ptr<IWICBitmapFrameEncode> frameEncoder;
    winrt::check_hresult(bitmapEncoder->CreateNewFrame(
        frameEncoder.put(), nullptr));
    winrt::check_hresult(frameEncoder->Initialize(nullptr));
    winrt::check_hresult(frameEncoder->WriteSource(
        wicBitmap, nullptr));
    winrt::check_hresult(frameEncoder->Commit());
    winrt::check_hresult(bitmapEncoder->Commit());

    // Rewind the stream so we can load it into a BitmapDecoder
    winrt::check_hresult(stream->Seek(
        { 0, 0 }, STREAM_SEEK_SET, nullptr));

    // Convert to a Windows Runtime RandomAccessStream.
    auto randomAccessStream = winrt::capture<
        winrt::IRandomAccessStream>(
        CreateRandomAccessStreamOverStream, stream.get(),
        BSOS_DEFAULT);

    // Convert the stream to a SoftwareBitmap.
    auto decoder = co_await winrt::BitmapDecoder::CreateAsync(
        randomAccessStream);
    co_return co_await decoder.GetSoftwareBitmapAsync();
}
```

This is very straightforward, but quite cumbersome. And you might feel a littly icky that you're encoding the bitmap as a PNG, only to immediately decode it back into a bitmap. Plus there's the complication that the Windows Runtime `BitmapDecoder` decodes asynchronously, so you have to do coroutine stuff to get the answer out.

Fortunately, there's a better way. We'll look at it next time.