How can I convert a Windows Runtime SoftwareBitmap to a WIC bitmap?

devblogs.microsoft.com/oldnewthing/20231122-00

November 22, 2023

Raymond Chen

Some time ago, I concluded a series of articles on how to convert a WIC bitmap to a Windows Runtime `SoftwareBitmap`. What about the reverse?

As I hinted in the Bonus Chatter, you can use the `ISoftwareBitmapNative` interface to reach inside the `SoftwareBitmap` and access the `IWICBitmap` hiding inside, if it has one.

```
#include <windows.graphics.imaging.interop.h>
#include <winrt/Windows.Graphics.Imaging.h>

winrt::com_ptr<IWICBitmap>
    GetWICBitmap(winrt::Windows::Graphics::Imaging::SoftwareBitmap const& bitmap)
{
    winrt::com_ptr wicBitmap;
    bitmap.as<ISoftwareBitmapNative>()->GetData(IID_PPV_ARGS(wicBitmap.put()));
    return wicBitmap;
}
```

Alternatively, if you like one-liners:

```
winrt::com_ptr<IWICBitmap>
    GetWICBitmap(winrt::Windows::Graphics::Imaging::SoftwareBitmap const& bitmap)
{
    return winrt::try_capture<IWICBitmap>(
        bitmap.as<ISoftwareBitmapNative>(),
        &ISoftwareBitmapNative::GetData);
}
```

The problem is that if the `SoftwareBitmap` is in a video format, the wrapped bitmap will be a `IMF2DBuffer`, not a `IWICBitmap`, and the call to `GetData` fails. In that case, we can convert the `SoftwareBitmap` to a WIC format and try again.

```
namespace winrt
{
    using namespace winrt::Windows::Graphics::Imaging;
}

winrt::com_ptr<IWICBitmap>
    GetWICBitmap(winrt::SoftwareBitmap const& bitmap)
{
    auto result = winrt::try_capture<IWICBitmap>(
        bitmap.as<ISoftwareBitmapNative>(),
        &ISoftwareBitmapNative::GetData);
    if (!result) {
        auto converted = winrt::SoftwareBitmap::
            Convert(bitmap, BitmapPixelFormat::Rgba8);

        result = winrt::try_capture<IWICBitmap>(
            converted.as<ISoftwareBitmapNative>(),
            &ISoftwareBitmapNative::GetData);
    }
    return result;
}
```

Note that in this second case, we are returning a copy of the pixels, since we did a conversion from the original format.

Now, maybe you want the `IWICBitmap` to have a specific format. In that case, you could force a conversion if the original `SoftwareBitmap` is not to your liking.

```
winrt::com_ptr<IWICBitmap>
    GetWICBitmapInFormat(
        winrt::SoftwareBitmap const& bitmap,
        winrt::BitmapPixelFormat format,
        winrt::BitmapAlphaMode alphaMode)
{
    winrt::SoftwareBitmap converted{ nullptr };
    if (bitmap.BitmapPixelFormat() == format &&
        bitmap.BitmapAlphaMode() == alphaMode) {
        converted = bitmap;
    } else {
        converted = winrt::SoftwareBitmap::Convert(bitmap, format, alphaMode);
    }
    return winrt::capture<IWICBitmap>(
            converted.as<ISoftwareBitmapNative>(),
            &ISoftwareBitmapNative::GetData);
}
```

If the `SoftwareBitmap` is already in the desired format, then you will get an `IWICBitmap` that shares pixels with the source. Otherwise, you get a private `IWICBitmap` whose pixels you are free to modify. If you'd rather get a private one all the time, you could force the conversion unconditionally.

```
winrt::com_ptr<IWICBitmap>
    GetCopyAsWICBitmap(
        winrt::SoftwareBitmap const& bitmap,
        winrt::BitmapPixelFormat format,
        winrt::BitmapAlphaMode alphaMode)
{
    auto converted = winrt::SoftwareBitmap::Convert(bitmap, format, alphaMode);
    return winrt::capture<IWICBitmap>(
            converted.as<ISoftwareBitmapNative>(),
            &ISoftwareBitmapNative::GetData);
}
```