

Obtaining attributed network usage information from the Windows Runtime

 devblogs.microsoft.com/oldnewthing/20210521-00

May 21, 2021



Raymond Chen

The network usage information we obtained last time can also be attributed, so you can see who is consuming all of your bandwidth.

Modify the C# application to gather data differently:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Windows.Networking.Connectivity;

class Program
{
    static async Task DoIt()
    {
        var now = DateTime.Now;
        var states = new NetworkUsageStates
        { Roaming = TriStates.DoNotCare, Shared = TriStates.DoNotCare };

        var profile = NetworkInformation.GetInternetConnectionProfile();
        Console.WriteLine($"Profile: {profile.ProfileName}");
        Console.WriteLine($"-----");

        var usages = await profile.GetAttributedNetworkUsageAsync(
            now.AddHours(-24), now, states);
        for (var usage in usages.OrderByDescending(u => u.BytesReceived +
u.BytesSent))
        {
            Console.WriteLine($"Id = {usage.AttributionId}");
            Console.WriteLine($"BytesReceived = {usage.BytesReceived}");
            Console.WriteLine($"BytesSent = {usage.BytesSent}");
            Console.WriteLine($"-----");
        }
    }

    static void Main()
    {
        DoIt().GetAwaiter().GetResult();
    }
}

```

This time, we get the current connection that is being used to access the Internet and ask for attributed network usage for the past 24 hours.

We sort the results descending by total byte transmitted (sent plus received) and print each record. The `AttributionId` identifies the source of the network access. It's kind of ugly, but you can usually eyeball it to see what the application is. (Converting this to something prettier is out of scope for this exercise.)

Here's the C++/WinRT version:

```

#include <winrt/Windows.Foundation.h>
#include <winrt/Windows.Foundation.Collections.h>
#include <winrt/Windows.Networking.Connectivity.h>
#include <stdio.h>
#include <algorithm>

using namespace std::literals::chrono_literals;
using namespace winrt;
using namespace winrt::Windows::Foundation;
using namespace winrt::Windows::Networking::Connectivity;

IAsyncAction DoIt()
{
    auto now = clock::now();
    NetworkUsageStates states{ TriStates::DoNotCare, TriStates::DoNotCare };

    auto profile = NetworkInformation::GetInternetConnectionProfile();
    printf("%ls\n", profile.ProfileName().c_str());
    printf("-----\n");

    auto usages = co_await profile.GetAttributedNetworkUsageAsync(
        now - 24h, now, states);
    std::vector<AttributedNetworkUsage> sorted{ begin(usages), end(usages) };
    std::sort(begin(sorted), end(sorted), [](auto&& left, auto&& right)
    {
        return left.BytesReceived() + left.BytesSent() >
            right.BytesReceived() + right.BytesSent();
    });
    for (auto usage : sorted) {
        printf("Id = %ls\n", usage.AttributionId().c_str());
        printf("BytesReceived = %I64u\n", usage.BytesReceived());
        printf("BytesSent = %I64u\n", usage.BytesSent());
        printf("-----\n");
    }
}

int main()
{
    init_apartment();
    DoIt().get();
}

```

The hardest part about translating the C# version to C++ was sorting the results descending by total bytes transferred!

Now you can answer this question that came in via a customer's liaison:

My customer wants to use the data usage feature built into Windows 10 and export the data to a CSV. We couldn't find a way to get this information from WMI, but it's there in the Settings app. How can we get that information?

We pointed the customer to the `NetworkInformation` class and the `AttributedNetworkUsage` class in particular.

| Can we use that class to export the data to a CSV?

The customer doesn't seem to want to do any actual programming. It looks they're hoping somebody has already written the program for them. Your exercise is to write that program.

Raymond Chen

Follow

