**Configuring .NET Network Tracing**  
From <https://blogs.msdn.microsoft.com/dgorti/2005/09/18/using-system-net-tracing/> :  
*“In Whidbey (.NET Framework 2.0) System.Net has a new feature called Tracing. This uses the builtin CLR Tracing functionality. What is interesting about the System.Net Tracing is that you can easily see the data that is being sent or received without having to use an app such as NETMON or Ethereal.”*   
**ME: Apparently, this system tracing also enables you to see SSL traffic and you don’t need to recompile your code!**

From <https://msdn.microsoft.com/en-us/library/a6sbz1dx(v=vs.110).aspx> :  
*“Network tracing provides access to information about method invocations* ***and network traffic*** *generated by a managed application.”*  
  
From <https://msdn.microsoft.com/en-us/library/hyb3xww8(v=vs.110).aspx> :  
*“To enable network tracing in the .NET Framework, you must select a destination for tracing output and add network tracing configuration settings to either the application or machine configuration file.”*  
**1. How to compile your code with tracing enabled.**   
**Note: As per my comment at the start of this document (and from my own testing), if you simply want to see the network traffic, you don’t need to recompile anything; you only need to create a logging config file (see steps below).**  
From <https://msdn.microsoft.com/en-us/library/64yxa344(v=vs.110).aspx> :

Using the command-line:  
For C#: **csc /d:TRACE MyApplication.cs**

**2. How to specify a log file for program tracing output.**From <https://msdn.microsoft.com/en-us/library/a6sbz1dx(v=vs.110).aspx> :

*To send traces to a log file, add the following node to the <system.diagnostics> node of the appropriate configuration file (application or machine). You can change the name of the file (trace.log) to suit your needs:*  
<system.diagnostics>

<trace autoflush="true" indentsize="4">

<listeners>

<add name="file" type="System.Diagnostics.TextWriterTraceListener" initializeData="**trace.log**"/>

</listeners>

</trace>

</system.diagnostics>

**Note:** If you don’t see a “trace.log” file being created, then it’s probably because your program is not making any trace API calls. However, if what you want to see is network traffic logging, then that will be in the “network.log” file (see below).  
  
**3. How to configure network tracing.**From <https://msdn.microsoft.com/en-us/library/ty48b824(v=vs.110).aspx> :

*The application* ***or*** *computer configuration file, holds the settings that determine the format and content of network traces.*

**computer configuration file method**The computer configuration file**, machine.config**, is stored in the **%Windir%\Microsoft.NET\Framework** folder in the directory where Windows was installed. There is a separate machine.config file in the folders under %Windir%\Microsoft.NET\Framework for each version of the .NET Framework installed on the computer (for example, C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\machine.config).  
  
**application configuration file method (MY PREFERRED METHOD)**This has precedence over the computer configuration file.   
Add the following lines to the appropriate configuration file (**Note**: The application configuration file needs to be named “<application-name>**.exe.config**” and be in the same directory as the executable.):  
  
<configuration>

<system.diagnostics>

<trace autoflush="true" indentsize="4">

<listeners>

<add name="file" type="System.Diagnostics.TextWriterTraceListener" initializeData="**trace.log**"/>

</listeners>

</trace>

<sources>

<source name="System.Net" tracemode="includehex" maxdatasize="1024">

<listeners>

<add name="System.Net"/>

</listeners>

</source>

<source name="System.Net.Cache">

<listeners>

<add name="System.Net"/>

</listeners>

</source>

<source name="System.Net.Http">

<listeners>

<add name="System.Net"/>

</listeners>

</source>

<source name="System.Net.Sockets">

<listeners>

<add name="System.Net"/>

</listeners>

</source>

<source name="System.Net.WebSockets">

<listeners>

<add name="System.Net"/>

</listeners>

</source>

</sources>

<switches>

<add name="System.Net" value="Verbose"/>

<add name="System.Net.Cache" value="Verbose"/>

<add name="System.Net.Http" value="Verbose"/>

<add name="System.Net.Sockets" value="Verbose"/>

<add name="System.Net.WebSockets" value="Verbose"/>

</switches>

<sharedListeners>

<add name="System.Net"

type="System.Diagnostics.TextWriterTraceListener"

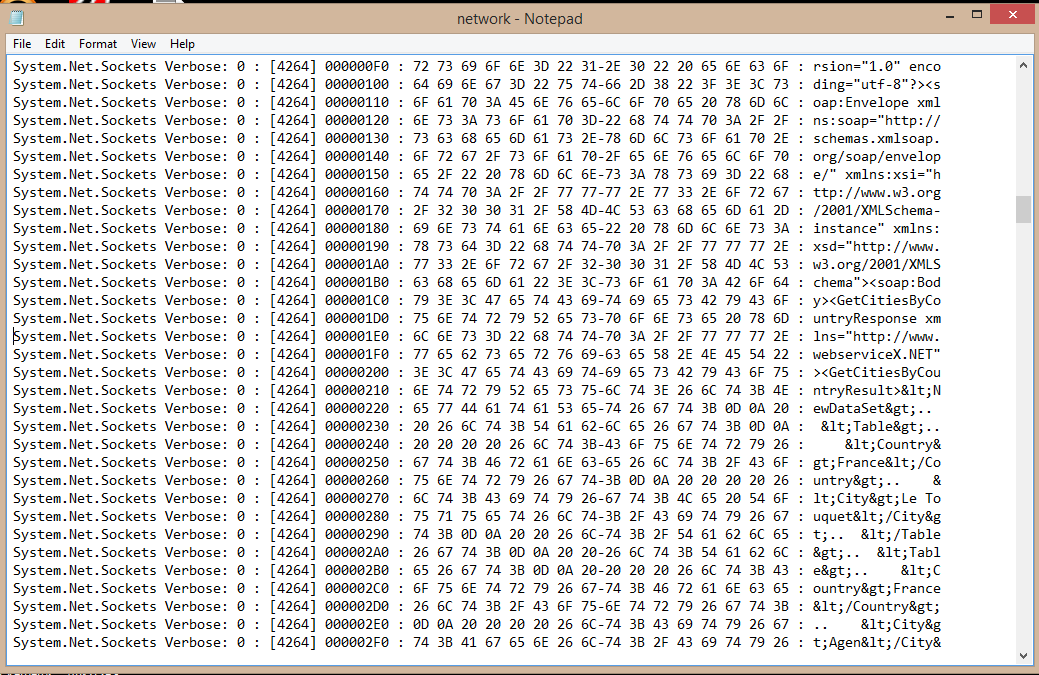
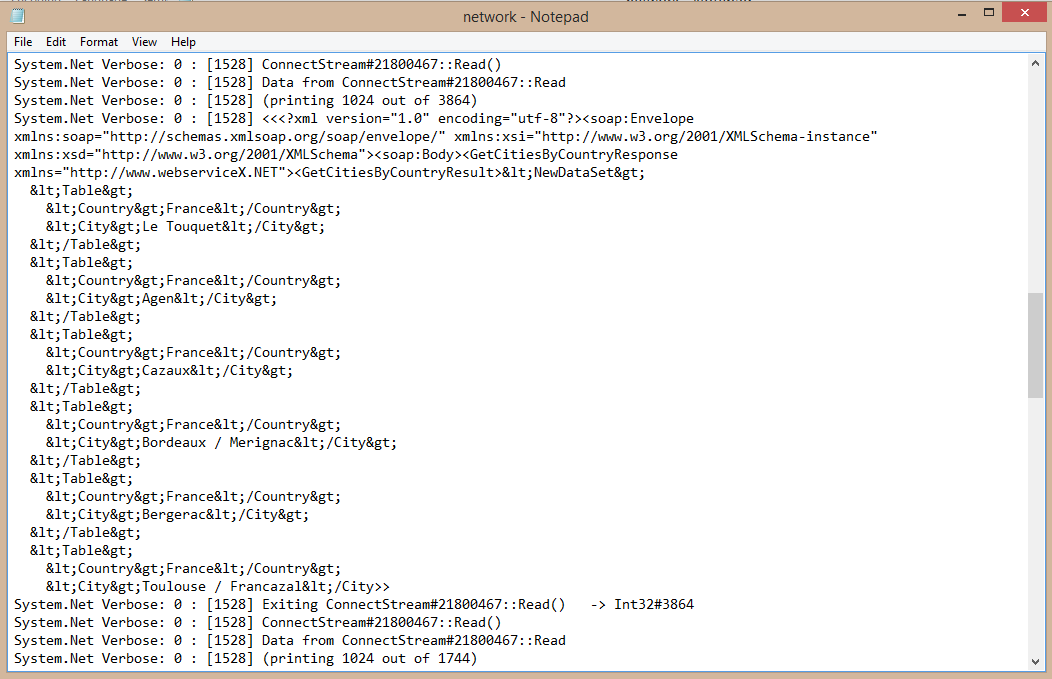
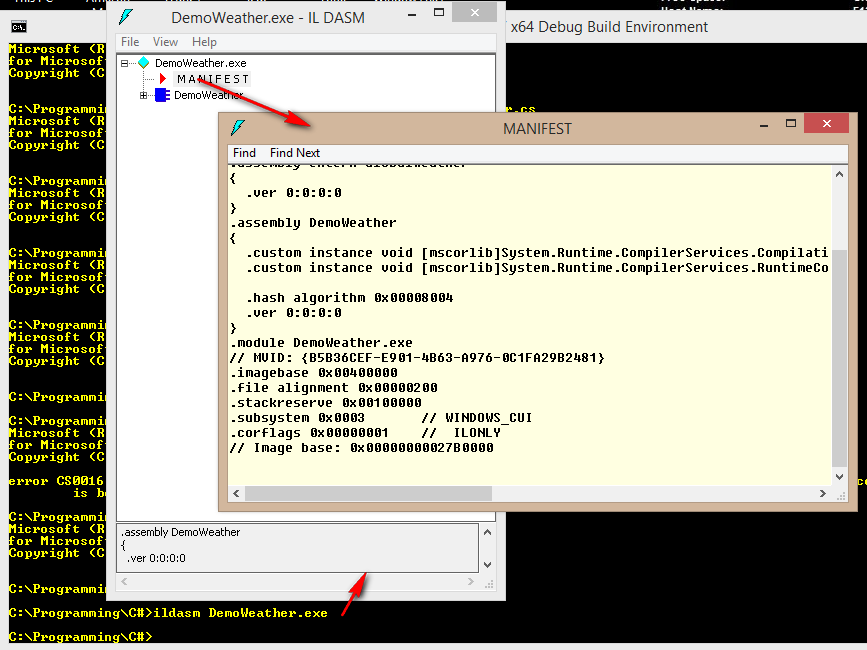
initializeData="**network.log**"

/>

</sharedListeners>

</system.diagnostics>

</configuration>  
  
**Demo: Response from web service to get list of cities in a specific country**C:\Programming\C#>DemoWeather.exe

Enter Country: France  
  
The “network.log” file included the following web service response:  
  
  
  
**Changing the config settings to control the logging output**Let’s change some of the logging settings (see bottom of this link for more details: <https://msdn.microsoft.com/en-us/library/ty48b824(v=vs.110).aspx)>:  
  
Restrict System.Net.Sockets tracing to Error:  
<add name="System.Net.Sockets" value="Error"/>  
  
Only print text (i.e. not text AND hex) from System.Net :  
<source name="System.Net" tracemode="protocolonly" maxdatasize="1024">  
  
Here’s how the same response looks now in the network.log file:  
  
**Note:** This makes the response XML a little easier to read.  
  
  
**Might be handy at some point: How to check the internals of a C# program using ILDASM**  
  
Interpreting .NET network traces:  
<https://msdn.microsoft.com/en-us/library/46fcs6sz(v=vs.110).aspx>  
  
*JeremyC 23/10/2016***END**