<https://www.codeproject.com/Articles/383162/Logging-and-Tracing-WCF-Soap-Messages>

## Introduction

Today, my team lead shoves me into an **R&D** to trace and log WCF soap messages. Then, I started reading different blog posts and books as well. After reading a couple of useful blogs, especially my favorite one **MSDN**,I came across the solutions that I want to share with you guys.

First of all, we know that, by default WCF does not record any messages. To initiate message logging, we have to add a trace listener to the System.ServiceModel.MessageLogging trace source and set attributes for the messageLogging element in the configuration file. To permit logging and tracing, we have to insert and stipulate some extra options in the formation.

The following example shows how to allow logging and specify essential options.

XML

Copy Code

<system.diagnostics>

<sources>

<source name="System.ServiceModel.MessageLogging">

<listeners>

<add name="ServiceModelMessageLoggingListener">

<filter type=""/>

</add>

</listeners>

</source>

</sources>

<sharedListeners>

<add initializeData="D:\SoapLog\Messages.svclog"

type="System.Diagnostics.XmlWriterTraceListener, System, Version=2.0.0.0,

Culture=neutral, PublicKeyToken=b77a5c561934e089"

name="ServiceModelMessageLoggingListener" traceOutputOptions="Timestamp">

<filter type=""/>

</add>

</sharedListeners>

</system.diagnostics>

In the example configuration, I have used sharedListeners which contains listeners but listeners in the sharedListeners collection are not placed in a Listeners collection. They must be added by name to the trace Element. In the example, the Listener name is"ServiceModelMessageLoggingListener**"** which is added to the sharedListeners collection and which adds the standard .NET Framework trace listener (System.Diagnostics.XmlWriterTraceListener) as the type to use. If we use System.Diagnostics.XmlWriterTraceListener, we have to stipulate the output file location and name in the configuration file. This is done by setting initializeData to the name of the log file. Otherwise, the system throws an exception. We might also implement a custom listener that releases logs to a default file.

WCF logs messages at two different levels, service and transport. Malformed messages are also logged. The three categories are self-governing from each other and can be triggered separately in configuration. We can control these levels by setting the logMessagesAtServiceLevel, logMalformedMessages, and logMessagesAtTransportLevel attributes of the messageLogging element.

So the messagelogging can look like below:

C#

Copy Code

<messageLogging

logEntireMessage="true"

logMalformedMessages="true"

logMessagesAtServiceLevel="false"

logMessagesAtTransportLevel="true"

maxMessagesToLog="2147483647"

maxSizeOfMessageToLog="2147483647"

/>

Like logging, tracing is also not supported by default. To initiate tracing, we have to form a trace listener and set a trace level for the trace source in configuration. Otherwise, WCF does not produce any traces. WCF outputs the following data for diagnostic tracing:

* Operation calls
* Code exceptions
* Warnings and other significant processing events
* Windows error events when the tracing feature malfunctions

As does logging, the following example shows how to allow tracing and specify essential options.

XML

Copy Code

<system.diagnostics>

<sources>

<source name="System.ServiceModel"

switchValue="Verbose,ActivityTracing" propagateActivity="true">

<listeners>

<add name="ServiceModelTraceListener">

<filter type=""/>

</add>

</listeners>

</source>

<sharedListeners>

<add initializeData="D:\SoapLog\Tracelog.svclog"

type="System.Diagnostics.XmlWriterTraceListener, System,

Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"

name="ServiceModelTraceListener" traceOutputOptions="Timestamp">

<filter type=""/>

</add>

</sharedListeners>

</system.diagnostics>

For my case, the configuration file looks like:

XML

Shrink ▲   Copy Code

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<system.serviceModel>

<diagnostics>

<messageLogging

logEntireMessage="true"

logMalformedMessages="true"

logMessagesAtServiceLevel="false"

logMessagesAtTransportLevel="true"

maxMessagesToLog="2147483647"

maxSizeOfMessageToLog="2147483647"

/>

</diagnostics>

<bindings>

<wsHttpBinding>

<binding name="WSHttpBinding\_IService" closeTimeout="00:01:00"

openTimeout="00:01:00"

receiveTimeout="00:10:00" sendTimeout="00:01:00"

bypassProxyOnLocal="false"

transactionFlow="false" hostNameComparisonMode="StrongWildcard"

maxBufferPoolSize="524288" maxReceivedMessageSize="65536"

messageEncoding="Text"

textEncoding="utf-8" useDefaultWebProxy="true"

allowCookies="false">

<readerQuotas maxDepth="32"

maxStringContentLength="8192" maxArrayLength="16384"

maxBytesPerRead="4096" maxNameTableCharCount="16384" />

<reliableSession ordered="true" inactivityTimeout="00:10:00"

enabled="false" />

<security mode="Message">

<transport clientCredentialType="Windows" proxyCredentialType="None"

realm="" />

<message clientCredentialType="Windows" negotiateServiceCredential="true"

algorithmSuite="Default" />

</security>

</binding>

</wsHttpBinding>

</bindings>

<client>

<endpoint address="http://localhost:8732/

Design\_Time\_Addresses/Rashim.RnD.TracingSoapMessage.Services/Service/"

binding="wsHttpBinding" bindingConfiguration="WSHttpBinding\_IService"

contract="ServiceReference.IService" name="WSHttpBinding\_IService">

<identity>

<dns value="localhost" />

</identity>

</endpoint>

</client>

</system.serviceModel>

<system.diagnostics>

<sources>

<source name="System.ServiceModel"

switchValue="Verbose,ActivityTracing" propagateActivity="true">

<listeners>

<add name="ServiceModelTraceListener">

<filter type=""/>

</add>

</listeners>

</source>

<source name="System.ServiceModel.MessageLogging">

<listeners>

<add name="ServiceModelMessageLoggingListener">

<filter type=""/>

</add>

</listeners>

</source>

</sources>

<sharedListeners>

<add initializeData="D:\SoapLog\Tracelog.svclog"

type="System.Diagnostics.XmlWriterTraceListener, System,

Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"

name="ServiceModelTraceListener" traceOutputOptions="Timestamp">

<filter type=""/>

</add>

<add initializeData="D:\SoapLog\Messages.svclog"

type="System.Diagnostics.XmlWriterTraceListener, System,

Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"

name="ServiceModelMessageLoggingListener" traceOutputOptions="Timestamp">

<filter type=""/>

</add>

</sharedListeners>

<trace autoflush="true"/>

</system.diagnostics>

</configuration>

This article was originally posted at <http://rashimuddin.wordpress.com/2012/05/11/how-to-log-and-trace-wcf-soap-messages>