.NET WCF TCP

This topic covers using .NET WCF TCP with SOAtest.

Sections include:

- Configuring .NET WCF TCP for Use in SOAtest
- Configuring Global .NET System Properties in SOAtest

Configuring .NET WCF TCP for Use in SOAtest

The **.NET WCF TCP** transport option allows you to invoke Windows Communication Foundation (.NET 3.0, 3.5, or 4.0) web services that use the TCP transport. For 4.0, the .NET 4 CLR / client runtime must be installed for this service.

SOAtest is able to understand WCF's system-provided `NetTcpBinding` and custom bindings that use the `TcpTransportBindingElement`. Microsoft describes the `NetTcpBinding` as "a secure and optimized binding suitable for cross-machine communication between WCF applications." For more information, see the following:

- System-provided bindings
- Custom bindings
- NetTcpBinding
- TcpTransportBindingElement

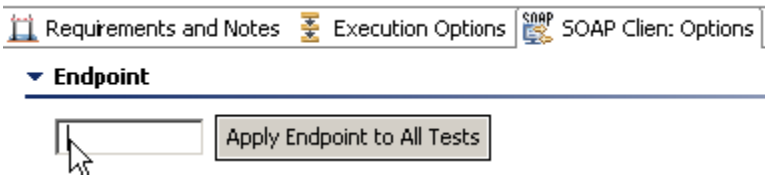
After selecting **.NET WCF TCP** from the **Transport** drop-down menu within the **Transport** tab of an applicable tool, the following options display in the left pane of the Transport tab:

- Endpoint
- Configuration File
- Security

Endpoint Options

The endpoint is the URL of the web service endpoint. By default, tool endpoints are set to the endpoint defined in the **WSDL**. The endpoint URL for WCF web services using the TCP transport begin with `"net.tcp://"` as opposed to `"http://"`. Besides WSDL, there are three other endpoint options:

- **Default:** When this option is selected, the endpoint will be the endpoint defined in the Test Suite that has the tool. To see the GUI for the endpoint defined in the Test Suite, double-click the Test Suite node and open the tool's Options tab:



- **Custom:** Allows you to set any custom endpoint.
- **UDDI serviceKey:** Describes what UDDI serviceKey is used to reference this server endpoint in the UDDI registry specified in the Preferences panel's WSDL/UDDI tab.

Configuration File Options

A WCF configuration file is an XML file that is used to define configuration settings for both WCF clients and services. The tool acts as a WCF client and can be configured using the settings from a WCF configuration file. If the Web service being tested has a WSDL document or metadata endpoint, a WCF client configuration file can be automatically generated by the Microsoft Service Model Metadata Tool. The Microsoft Service Configuration Editor can also be used to create and edit WCF configuration files using a graphical user interface. These tools are widely available as part of Microsoft Windows SDK.

Tools can automatically determine endpoint binding information based on the policies defined in the WSDL, or based on the endpoint binding configuration defined in a WCF client configuration file. When the **Constrain to WSDL** tool option is selected, the endpoint's binding is automatically determined based on the policies defined in the WSDL. When **Constrain to WSDL** in unchecked, SOAtest will look in the specified WCF client configuration file for the endpoint's binding settings.

Tools can also use the endpoint behavior information defined in the WCF client configuration file. An endpoint behavior section containing a `clientCredentials` element must be present if the service will be negotiating service credentials using certificates.

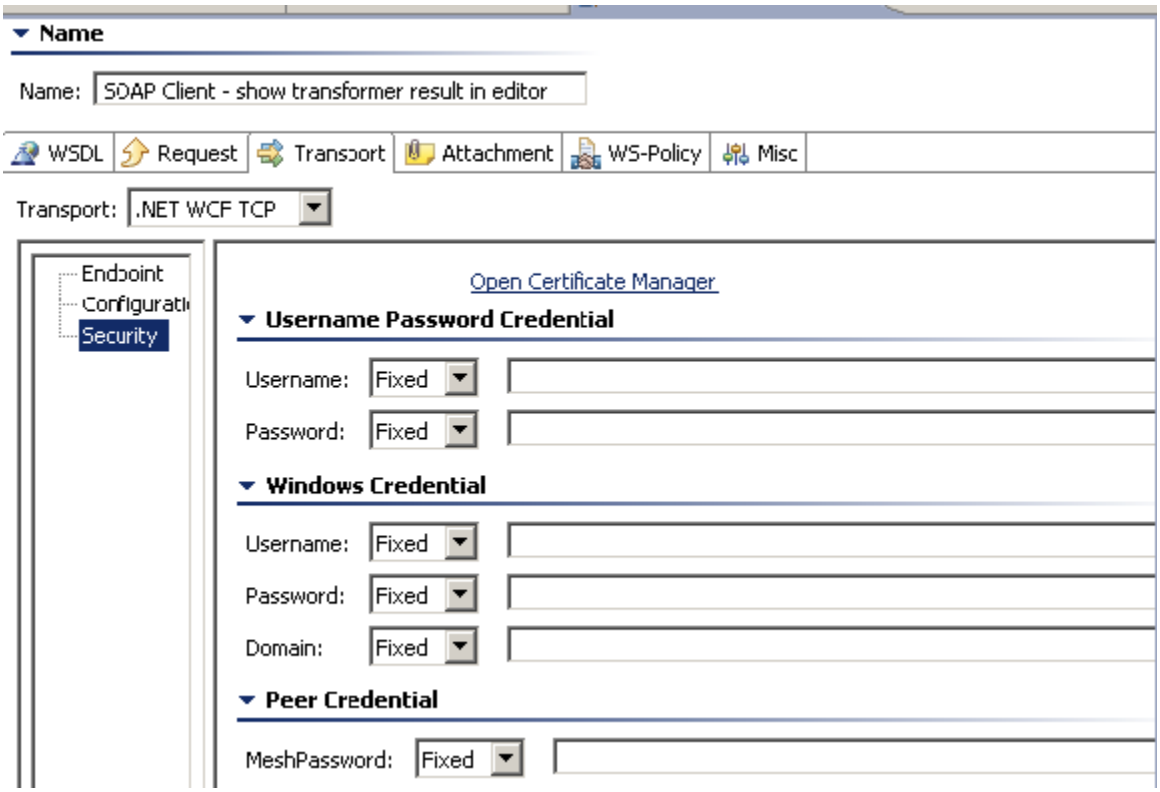
The following Configuration File options are available:

- **WCF Client Configuration File:** Click the **Browse** button to select the desired configuration file.
- **Persist As Relative Path:** Check this option if you want the path to this file to be saved as a path that is relative to the current configuration file. Enabling this option makes it easier to share tests across multiple machines. If this option is not enabled, the test suite will save the path to this file as an absolute path.

- **Open Service Configuration Editor:** This option is only displayed if Microsoft's Service Configuration Editor is installed on your machine. This feature, allows you to quickly open the editor to create a new service configuration file, or to view or edit the currently selected service configuration file.

Security Options

The Security options tab allows you to enter username and password information for various security credentials. In addition, the **Open Certificate Manager** option is available. Click **Open Certificate Manager** to open the Windows Certificate Manager. The Windows Certificate Manager allows you to manage any certificates needed for authentication when invoking your Web service. The WCF Configuration file may have references to certificates shown by the Windows Certificate Manager



Configuring the Request

From the **Request** tab, you can configure the SOAP envelope for the request message. This SOAP envelope is configured before any transport and message transformations, such as encryption. .NET WCF will automatically apply the message and transport security when you run the test. The XML Signer and XML Encryption tools are not necessary and should not be chained to the tool. Also, no security headers need to be added to the SOAP request. The only SOAP header that should be present is a WS-Addressing header which SOAtest automatically adds if your test was created from a WSDL.

Configuring Transaction Support

SOAtest supports flowed transactions via the WS-Atomic Transactions protocol and MS OLE Transactions protocol for WCF web services. For more information, see .NET WCF Flowed Transactions.

Configuring Global .NET System Properties in SOAtest

Global properties in the .NET System.Net.ServicePointManager class can be configured by setting java system properties on the SOAtest command line. The ServicePointManager.DefaultConnectionLimit limit property limits the number of HTTP keep alive connections between SOAtest and a remote host.

The default value for the DefaultConnectionLimit property is 2. If you are performing load testing and want more than two HTTP keep-alive connections, we recommend increasing this property.

For example, to set the DefaultConnectionLimit property to 50 you can pass "-J-DSystem.Net.ServicePointManager.DefaultConnectionLimit=50" as a command line argument when starting SOAtest. The full list of supported ServicePointManager properties that can be configured on the SOAtest command line are listed below:

- System.Net.ServicePointManager.CheckCertificateRevocationList

- System.Net.ServicePointManager.DefaultConnectionLimit
- System.Net.ServicePointManager.DnsRefreshTimeout
- System.Net.ServicePointManager.EnableDnsRoundRobin
- System.Net.ServicePointManager.Expect100Continue
- System.Net.ServicePointManager.MaxServicePointIdleTime
- System.Net.ServicePointManager.MaxServicePoints
- System.Net.ServicePointManager.SecurityProtocol
- System.Net.ServicePointManager.UseNagleAlgorithm

For more information about these properties please see http://msdn.microsoft.com/en-us/library/system.net.servicepointmanager_properties.aspx .