

# .NET WCF SOAP Clients - FAQ

This topic answers common questions regarding .NET WCF SOAP Clients.

Sections include:

- [What is a binding?](#)
- [How do I create a .NET WCF client configuration file?](#)
- [Do I have to use a .NET WCF client configuration file?](#)
- [How do I configure certificate-based authentication?](#)
- [Can I use a .NET WCF SOAP Client to test older .NET web services?](#)
- [Can I use a .NET WCF SOAP Client to test Glassfish Metro/WSIT based services?](#)
- [Common Errors](#)

## What is a binding?

In WCF, a binding describes the transport, security, and message encoding for a web service's endpoint. The SOAP Client needs to know the endpoint's binding in order to communicate with the endpoint. The SOAP Client can determine the endpoint's binding automatically from a WSDL document or from a .NET WCF client configuration file.

## How do I create a .NET WCF client configuration file?

The Microsoft Service Model Metadata Tool (svcutil.exe) can generate a client configuration file given the URL of the service's WSDL document. The Microsoft Service Configuration Editor can be used to create and edit WCF configuration files using a graphical user interface. Both of these tools are part of the Microsoft Windows SDK, which is included with Visual Studio and can also be downloaded from Microsoft. The web service's developer may also be able to provide you with a client configuration file for the web service that you are testing.

## Do I have to use a .NET WCF client configuration file?

Not necessarily. If the web service's WSDL document sufficiently describes the web service, then the SOAP Client only needs to be constrained to that WSDL without using a client configuration file. However, if the web service uses certificate-based authentication, then you must setup a .NET WCF client configuration file that describes the locations of the certificates installed on your local machine. WCF bindings also define various limits, including a limit on the maximum received message size. You can control such limits in a .NET Client configuration file. You also need to un-constrain your SOAP Client test from the WSDL document to prevent the binding information in the WSDL from taking precedence over what you have in your configuration file.

## How do I configure certificate-based authentication?

First, you need to install any client and server certificates using the Windows Certificate Manager. You can open the Windows Certificate Manger by clicking **Start> Run...**, entering "`certmgr.msc`", then clicking **OK**. Typically, you would install the server's certificate in the Trusted People store and any client certificates in the Personal store. Next, the locations of those certificates must be specified in your client configuration file.

The MSDN web site has detailed information about how to do this, including configuration file examples:

- Working with Certificates: <http://msdn.microsoft.com/en-us/library/ms731899.aspx>
- How to: Specify Client Credential Values: <http://msdn.microsoft.com/en-us/library/ms732391.aspx>
- Transport Security with Certificate Authentication: <http://msdn.microsoft.com/en-us/library/ms731074.aspx>
- Message Security with a Certificate Client: <http://msdn.microsoft.com/en-us/library/ms733098.aspx>

## Can I use a .NET WCF SOAP Client to test older .NET web services?

Only WSE 3.0. .NET WCF clients are wire-level compatible with WSE 3.0 services. The following custom binding can be used to test a WSE 3.0 web service using Secure Conversation and MutualCertificate11:

```

<binding name="SecureConversationMutualCertificate11SignEncryptConfig">
  <security authenticationMode="SecureConversation" messageProtectionOrder=
    "SignBeforeEncrypt" requireSecurityContextCancellation="false">
    <secureConversationBootstrap authenticationMode="MutualCertificate"
      securityHeaderLayout="Strict" messageProtectionOrder="SignBeforeEncrypt"
      requireSignatureConfirmation="true" />
  </security>
  <textMessageEncoding messageVersion="Soap11WSAddressingAugust2004" />
  <httpTransport />
</binding>

```

The MSDN web site has more detail; see <http://msdn2.microsoft.com/en-us/library/ms730299.aspx> .

## Can I use a .NET WCF SOAP Client to test Glassfish Metro/WSIT based services?

Yes! Metro WSIT services are interoperable with .NET WCF clients, including SOAtest's SOAP Client tool. However, there is a known bug in Metro where whitespace and newlines cause the security validation for the request message to fail. This problem can be avoided by chaining an Extension tool to the Request SOAP Envelope output of your SOAP Client test that to remove the extra white space:

```

from com.parasoft.api import *
from java.lang import *

def removeAllNewLineCharacters(input, context):
    noNewLines = str(input).replace("\r", "").replace("\n", "")
    result = String(noNewLines).replaceAll(">\s*<", "><")
    Application.showMessage(result) # for debugging
    return result

```

## Common Errors

Error Message	Resolution
Could not find endpoint in client configuration file	This test failure will happen if your SOAP Client test is using a .NET WCF client configuration file that does not describe the endpoint configured for your test. Verify that the client configuration file describes the same endpoint as the test. Check the capitalization of each character in the endpoint URL (case-sensitive). Also verify that your test is using a valid .NET WCF client configuration file. A service configuration file (web.config) or WSE configuration file will not work.
Could not find endpoint in WSDL	This test failure will happen if your SOAP Client test is constrained to a WSDL document that does not describe the endpoint configured for your test. You may need to correct the endpoint in your test to match the endpoint in the WSDL document. If the endpoint in the WSDL document is incorrect, then the WSDL should be corrected. If you cannot correct the WSDL, then you should un-constrain the SOAP Client test from the bad WSDL then configure your test to use a .NET WCF client configuration file that describes the correct endpoint.
The X.509 certificate CN=localhost chain building failed. The certificate that was used has a trust chain that cannot be verified. Replace the certificate or change the certificateValidationMode.	As the test failure suggests, you likely need to configure the certificateValidationMode property in your client configuration file. Typically, you would add the server's certificate to the Trusted People store using the Windows certificate manager then set the certificateValidationMode property in your configuration file to PeerOrChainTrust. The MSDN web site has an example; see <a href="http://msdn.microsoft.com/en-us/library/ms587504.aspx">http://msdn.microsoft.com/en-us/library/ms587504.aspx</a> and <a href="http://msdn.microsoft.com/en-us/library/system.servicemodel.security.x509certificatevalidationmode.aspx">http://msdn.microsoft.com/en-us/library/system.servicemodel.security.x509certificatevalidationmode.aspx</a> .
Identity check failed for outgoing message. The expected DNS identity of the remote endpoint was 'host1' but the remote endpoint provided DNS clam 'host2'. If this is a legitimate remote endpoint, you can fix the problem by explicitly specifying DNS identity 'host2' as the Identity property of EndpointAddress when creating channel proxy.	The endpoint's DNS identity in the WSDL may be incorrect. As the test failure suggests, you likely need to explicitly configure the Identity property in your client configuration file. You will also need to un-constrain your test from the WSDL to prevent the endpoint information in the WSDL from taking precedence over what you have in your configuration file.  See specifying Identity at the Client: <a href="http://msdn.microsoft.com/en-us/library/ms733130.aspx">http://msdn.microsoft.com/en-us/library/ms733130.aspx</a> .

<p>The maximum message size quota for incoming messages (65536) has been exceeded. To increase the quota, use the MaxReceivedMessageSize property on the appropriate binding element."</p>	<p>As the test failure suggests, you need to configure the MaxReceivedMessageSize property in your client configuration file. You will also need to un-constrain your test from the WSDL to prevent the binding information in the WSDL from taking precedence over what you have in your configuration file. The MSDN web site has an example; see <a href="http://msdn.microsoft.com/en-us/library/system.servicemodel.basichttpbinding.maxreceivedmessagesize.aspx">http://msdn.microsoft.com/en-us/library/system.servicemodel.basichttpbinding.maxreceivedmessagesize.aspx</a> .</p>
<p>The maximum read depth (32) has been exceeded because XML data being read has more levels of nesting than is allowed by the quota. This quota may be increased by changing the MaxDepth property on the XmlDictionaryReaderQuotas object used when creating the XML reader.</p>	<p>As the test failure suggests, you need to configure the MaxDepth property in your client configuration file. You will also need to un-constrain your test from the WSDL to prevent the binding information in the WSDL from taking precedence over what you have in your configuration file. The MSDN web site has an example; see <a href="http://msdn.microsoft.com/en-us/library/system.servicemodel.basichttpbinding.readerquotas.aspx">http://msdn.microsoft.com/en-us/library/system.servicemodel.basichttpbinding.readerquotas.aspx</a> .</p>
<p>"Out of memory" errors when load testing .NET WCF SOAP Clients.</p>	<p>You may need to reduce the size of the Java heap so that the .NET runtime can allocate more memory for the .NET heap.</p>
<p>Unrecognized message version</p>	<p>This error is referring to the version of SOAP and WS-Addressing of the message being sent. This test failure can occur if the outgoing message is not a valid SOAP envelope. If the SOAP Client's request tab is using the Literal XML or Form XML view, verify that a SOAP envelope is present in the XML.</p>