



## Managing WCF Services with Policy Manager Guide

## ***Trademarks***

SOA Software and the SOA Software logo are either trademarks or registered trademarks of SOA Software, Inc. Other product names, logos, designs, titles, words or phrases mentioned within this guide may be trademarks, service marks or trade names of SOA Software, Inc. or other third parties and may be registered in the U.S. or other jurisdictions.

## ***Copyright***

©2001-2010 SOA Software, Inc. All rights reserved. No material in this manual may be copied, reproduced, republished, uploaded, posted, transmitted, distributed or converted to any electronic or machine-readable form in whole or in part without prior written approval from SOA Software, Inc.

# Table of Contents

<b>MANAGING WCF SERVICES WITH POLICY MANAGER GUIDE .....</b>	<b>I</b>
<b>PREFACE .....</b>	<b>4</b>
What's in this Guide? .....	4
Other Documentation .....	4
Customer Support .....	5
<b>CHAPTER 1: AGENT FOR WCF ARCHITECTURE .....</b>	<b>6</b>
Overview .....	6
Architecture .....	6
<b>CHAPTER 2: MANAGING WCF SERVICES WITH POLICY MANAGER SECURITY POLICIES .</b>	<b>9</b>
Overview .....	9
AnonymousForCertificate .....	9
MutualCertificateSignEncrypt and MutualCertificateSignOnly .....	10
MutualCertificateSymmetricBinding .....	11
UsernameForCertificate .....	12
UsernameOverTransport .....	13
CertificateOverTransport .....	14
KerberosOverTransport .....	15
SAMLOverTransport .....	16
Configuring SOA Software WS-Auditing Policies .....	16
Usage Data Monitoring .....	24
<b>CHAPTER 3: POLICY MANAGER CONFIGURATIONS FOR WCF COMMON SECURITY SCENARIOS .....</b>	<b>25</b>
Overview .....	25
Configure Services .....	25
Transport Security with Basic Authentication .....	26
Message Security with a Windows Client over HTTP .....	29
Message Security with a Windows Client over NET.TCP .....	33
Message Security with a User Name Client .....	36
Transport Security with Windows Authentication over HTTPS .....	41
Message Security with a Certificate Client .....	44
Configuring Other WCF Scenarios .....	48
<b>CHAPTER 4: CONFIGURING WCF POLICIES WITH NETWORK DIRECTOR .....</b>	<b>49</b>
Overview .....	49
Configuring Virtual Service Inbound Policies .....	49
Configuring Virtual Service Outbound Policies .....	53

# Preface

## WHAT'S IN THIS GUIDE?

The Managing WCF Services with Policy Manager Guide provides an overview of the "Service Manager Agent for WCF" architecture, and information on policy configuration and common security scenarios.

It includes the following chapters:

- Chapter 1, "Agent for WCF Architecture," provides an overview of the Agent for WCF solution and its three major components: soaBinding, SOA Agent Windows Service, and Centralized Internal Metabase.
- Chapter 2, "Managing WCF Services with Policy Manager Security Policies," describes the "Operational" sample policies that are supported by the Agent for WCF and provides configuration details.
- Chapter 3, "Policy Manager Configurations for WCF Common Security Scenarios," describes typical Intranet and Internet scenarios that are described by the variations of Microsoft WCF wsHttpBinding and netTcpBinding configurations.
- Chapter 4, "Configuring WCF Policies with Network Director," describes typical policy configurations supported by Policy Manager when WCF services are virtualized through Network Director.

## OTHER DOCUMENTATION

To effectively use this guide, you should have access to and a working knowledge of the concepts outlined in the following Policy Manager product documentation:

- Policy Manager 6.0 Installation Guide for Windows and UNIX Platforms
- Policy Manager Online Help
- Service Manager Agent for WCF Installation Guide
- Microsoft Visual Studio Add-in Users Guide

## CUSTOMER SUPPORT

SOA Software offers a variety of support services to our customers. The following options are available:

<b>Support Options:</b>	
Email (direct)	<a href="mailto:support@soa.com">support@soa.com</a>
Phone	1-866 SOA-9876 (1-866-762-9876)
Email (Web)	The "Support" section of the SOA Software website ( <a href="http://www.soa.com">www.soa.com</a> ) provides an option for emailing product related inquiries to our support team.
Documentation Updates	Updates to Policy Manager product documentation are issued on a monthly basis and are available by submitting an email request to <a href="mailto:support@soa.com">support@soa.com</a> .

# Chapter 1: Agent for WCF Architecture

## OVERVIEW

The "Service Manager Agent for WCF" is built on top of Microsoft Windows Communication Foundation (WCF) technology and fully leverages Microsoft .NET Framework and WCF.

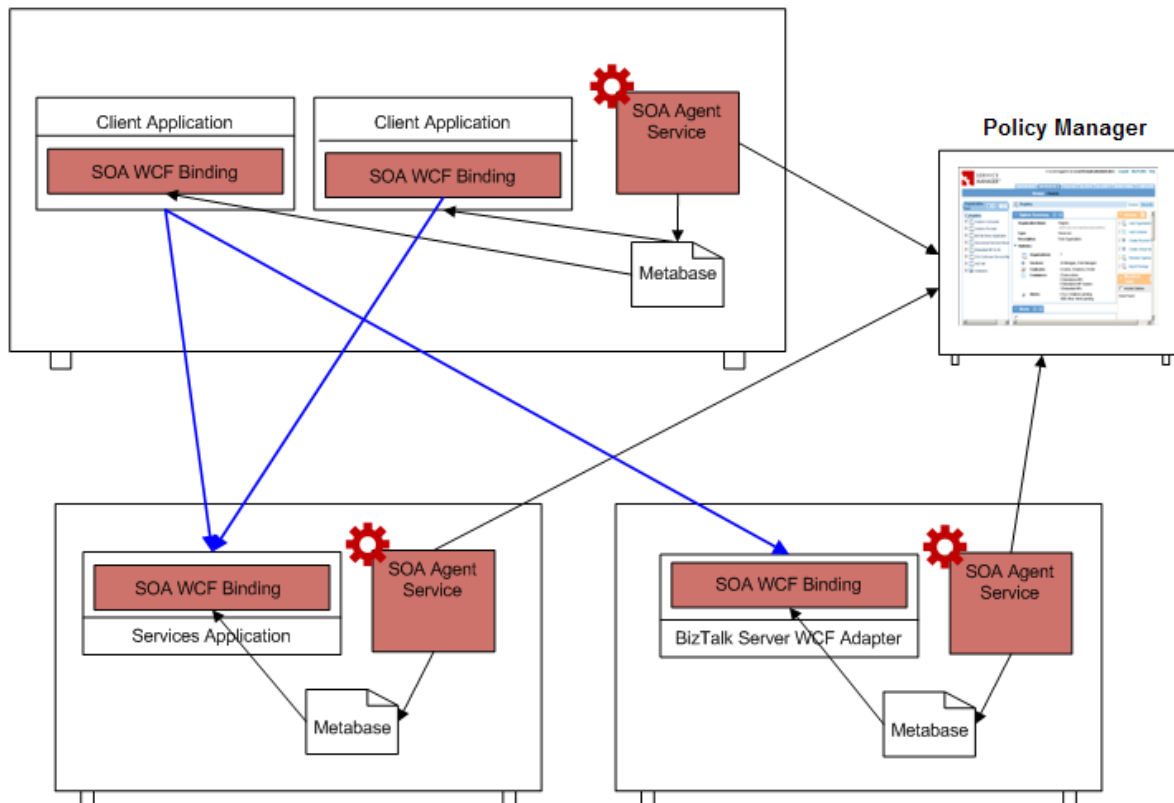
When services are enabled with SOA Software manageability they remain native WCF services. The Agent for WCF only drives the underlying WCF mechanism to dynamically enforce and/or implement security and standard interoperable and WCF specific policies. Additionally the Agent for WCF extends WCF with functionality not readily available out-of-the-box by Microsoft WCF (e.g., services monitoring, centralized messages recording, auditing, declarative authorization, etc.).

As a native WCF solution, the Agent for WCF is built using extensibility points of the WCF technology. Specifically, the Agent for WCF is leveraging the WCF notion of extensible WCF channels that are defined by WCF binding(s).

## ARCHITECTURE

At the core of the Agent for WCF solution is a custom developed `soaBinding`, and a new WCF binding configuration element that is introduced to the WCF design and runtime as soon as the Agent is installed. The fundamental difference between an `soaBinding` and any other standard WCF binding is that an `soaBinding` is not configured with specific policies, text encoders or transport channel attributes, but instead carries only short links to a service endpoint registered in the SOA Software UDDI v3 Registry.

When the WCF Service Host is started during runtime, the `soaBinding` dynamically builds a standard WCF channel stack according to the policies defined in UDDI Registry database. In most cases the `soaBinding` extends the WCF channel stack with additional channels provided by SOA Software, which implements functionality unique to the Agent for WCF product. The diagram below shows the Agent for WCF high-level architecture.



**Figure 1.1: Agent for WCF Architecture**

The Agent for WCF solution includes three major components:

1. soaBinding - A binding that replaces the standard WCF binding in application configuration files (ex., web.config).
2. SOA Agent Windows Service – A service that is responsible for communication with the Policy Manager instance(s) on behalf of all soaBinding's available on a given computer.
3. Centralized Internal Metabase – A persistent cache of the actual and current soaBinding configurations.

An soaBinding contains standard WCF binding functionality, and can be used with service endpoints or client endpoints. This symmetry results in two possible Agent container types that a single Agent for WCF installation can be configured with. These include Agent and Delegate container types. Agent containers manage service endpoints, and Delegate containers manage client endpoints. Each container is a logical grouping of either service or client WCF endpoints on a given computer.

Figure 1-1 illustrates a hypothetical setup where service application(s) are deployed in an IIS/WAS environment (bottom left computer) and a BizTalk Server environment (bottom right computer). The top left computer hosts client applications that call SOA Software managed services. Client applications are also shown as managed by the SOA Software soaBinding. It is

not a requirement to place both client and service applications under management. They can be managed independently, or not managed.

Figure 1-2 illustrates how the Agent for WCF leverages the WCF channel stack by placing and dynamically configuring standard WCF channels, as well as injecting custom built Intercepting channels.

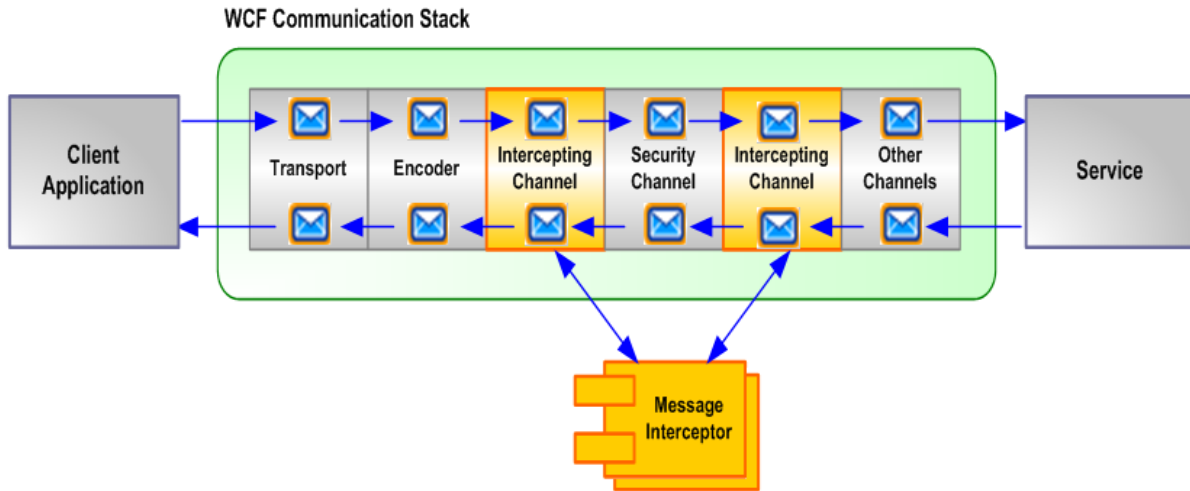


Figure 1-2: Agent for WCF Communication Stack



# Chapter 2: Managing WCF Services with Policy Manager Security Policies

## OVERVIEW

Policy Manager is shipped with a series of pre-defined sample policies that illustrate common interoperable security scenarios. These policies are located in the Root Organization of the Policy Manager "Management Console" and can be used as a base policy and customized to meet your requirements. Sample policies are typically constructed using a Policy Manager "Aggregate" policy and include two or more added or referenced policies.

This chapter describes the "Operational" sample policies that are supported by the Agent for WCF and provides configuration details. Each section provides the following information:

- A description of the Policy Manager sample policy (Aggregate and associated policies).
- Notation indicating if the policy is supported by the Agent for WCF.
- A WCF binding configuration that matches the Policy Manager sample policy.
- Additional information if the policy requires configuration beyond what is illustrated in the WCF binding configuration.

## ANONYMOUSFORCERTIFICATE

The AnonymousForCertificate policy is an Aggregate Policy that includes the following policies:

- AnonymousForCertificate\_part1 (WS-Security Symmetric Binding Policy)
- AnonymousForCertificate\_part2 (WS-Security Message Policy)

In this policy, the client is anonymous and the service is authenticated using an X.509 certificate. The WCF binding used is an instance of a Symmetric Binding. The SOAP body is signed and encrypted with the derived keys.

### Supported by Agent for WCF

Yes

### WCF Configuration

```
<customBinding>  
  <binding name="AnonymousForCertificate">  
    <security defaultAlgorithmSuite="TripleDesRsa15"  
authenticationMode="AnonymousForCertificate" />  
  </binding>  
</customBinding>
```

```

        requireDerivedKeys="true" securityHeaderLayout="Lax" includeTimestamp="true"
        keyEntropyMode="CombinedEntropy" requireSignatureConfirmation="false"
        messageProtectionOrder="SignBeforeEncryptAndEncryptSignature"
        messageSecurityVersion="WSSecurity11WSTrustFebruary2005WSecureConversationFebr
uary2005WSSecurityPolicy11BasicSecurityProfile10" />
        <textMessageEncoding messageVersion="Soap12" />
        <httpTransport authenticationScheme="Anonymous" />
    </binding>
</customBinding>

```

The AnonymousForCertificate policy requires that the client be configured with the service's X.509 certificate. This is accomplished using endpoint behavior:

```

<endpointBehaviors>
  <behavior name="ClientBehavior">
    <clientCredentials>
      <serviceCertificate>
        <defaultCertificate findValue="[certificate hash value]"
          storeLocation="LocalMachine" storeName="My"
          x509FindType="FindByThumbprint"/>
      </serviceCertificate>
    </clientCredentials>
  </behavior>
</endpointBehaviors>

```

## MUTUALCERTIFICATESIGNENCRYPT AND MUTUALCERTIFICATESIGNONLY

The MutualCertificateSignEncrypt policy is an Aggregate Policy that includes the following policies:

- MutualCertificateSignEncrypt\_part1 (WS-Security Asymmetric Binding Policy)
- MutualCertificateSignEncrypt\_part2 (WS-Security Message Policy)

In this policy, the client authenticates using an X.509 certificate which appears at the SOAP layer as the initiator token. The service is also authenticated using an X.509 certificate. The SOAP body will be signed and optionally encrypted.

The MutualCertificateSignOnly policy is an Aggregate Policy that includes the following policies:

- MutualCertificateSignOnly\_part1 (WS-Security Asymmetric Binding Policy)
- MutualCertificateSignOnly\_part2 (WS-Security Message Policy)

In this policy, the client authenticates using an X.509 certificate which appears at the SOAP layer as the initiator token. The service is also authenticated using an X.509 certificate. The SOAP body will be signed.

### Supported by Agent for WCF

Yes

## WCF Configuration

```
<customBinding>
  <binding name="MutualCertificateDuplex">
    <security defaultAlgorithmSuite="TripleDesRsa15"
authenticationMode="MutualCertificateDuplex"
    requireDerivedKeys="false" securityHeaderLayout="Lax" includeTimestamp="true"
    keyEntropyMode="CombinedEntropy" messageProtectionOrder="SignBeforeEncrypt"
    messageSecurityVersion="WSSecurity10WSTrustFebruary2005WSSecureConversationFebr
    uary2005WSSecurityPolicy11BasicSecurityProfile10"
    requireSignatureConfirmation="false" />
    <textMessageEncoding messageVersion="Soap12" />
    <httpTransport authenticationScheme="Anonymous" />
  </binding>
</customBinding>
```

The MutualCertificateSignEncrypt and MutualCertificateSignOnly policies require the client to be configured with the service's X.509 certificate. This is accomplished using endpoint behavior:

```
<endpointBehaviors>
  <behavior name="ClientBehavior">
    <clientCredentials>
      <serviceCertificate>
        <defaultCertificate findValue="[certificate hash value]"
          storeLocation="LocalMachine" storeName="My"
          x509FindType="FindByThumbprint" />
      </serviceCertificate>
    </clientCredentials>
  </behavior>
</endpointBehaviors>
```

## MUTUALCERTIFICATESYMMETRICBINDING

The MutualCertificateSymmetricBinding policy is an Aggregate Policy that includes the following policies:

- MutualCertificateSymmetricBinding\_part1 (WS-Security Symmetric Binding Policy)
- MutualCertificateSymmetricBinding\_part2 (WS-Security Message Policy)
- MutualCertificateSymmetricBinding\_part3 (WS-Security Supporting Tokens Policy)

In this policy, the client authenticates using an X.509 certificate as an endorsing supporting token. The service is also authenticated using an X.509 certificate. The binding used is a symmetric binding with the protection token being a key generated by the client, encrypted with the public key of the service. The SOAP body is signed and encrypted by the derived keys.

### Supported by the AGENT for WCF

Yes

## WCF Configuration

```
<customBinding>
  <binding name="MutualCertificate">
```

```

    <security defaultAlgorithmSuite="TripleDesRsa15"
authenticationMode="MutualCertificate"
    requireDerivedKeys="true" securityHeaderLayout="Lax" includeTimestamp="true"
    keyEntropyMode="CombinedEntropy"
messageProtectionOrder="SignBeforeEncryptAndEncryptSignature"
messageSecurityVersion="WSSecurity11WSTrustFebruary2005WSecureConversationFebr
uary2005WSecurityPolicy11BasicSecurityProfile10"
    requireSignatureConfirmation="false" />
    <textMessageEncoding messageVersion="Soap12" />
    <httpTransport authenticationScheme="Anonymous" />
</binding>
</customBinding>

```

The MutualCertificateSymmetricBinding policy requires that the client be configured with the service's X.509 certificate and its own certificate. This is accomplished using endpoint behavior:

```

<endpointBehaviors>
  <behavior name="ClientBehavior">
    <clientCredentials>
      <clientCertificate findValue="[certificate hash value]"
        storeLocation="LocalMachine" storeName="My"
        x509FindType="FindByThumbprint" />
      <serviceCertificate>
        <defaultCertificate findValue="e0202fe0253d76c587c52bc96811357d46a010da"
          storeLocation="LocalMachine" storeName="My"
          x509FindType="FindByThumbprint" />
      </serviceCertificate>
    </clientCredentials>
  </behavior>
</endpointBehaviors>

```

## USERNAMEFORCERTIFICATE

The UsernameForCertificate policy is an Aggregate Policy that includes the following policies:

- UsernameForCertificate\_part1 (WS-Security Symmetric Binding Policy)
- UsernameForCertificate\_part2 (WS-Security Message Policy)
- UsernameForCertificate\_part3 (WS-Security Supporting Tokens Policy)

In this policy, the client authenticates to the service using a Username Token which appears at the SOAP layer as a signed supporting token. The service authenticates to the client using an X.509 certificate. The binding used is a symmetric binding with the protection token being a key generated by the client, encrypted with the public key of the service. The SOAP body is signed and encrypted by the derived keys.

### Supported by the Agent for WCF

Yes

### WCF Configuration

```

<customBinding>
  <binding name="UserNameForCertificate">

```

```

    <security defaultAlgorithmSuite="TripleDesRsa15"
authenticationMode="UserNameForCertificate"
    requireDerivedKeys="true" securityHeaderLayout="Lax" includeTimestamp="true"
    keyEntropyMode="CombinedEntropy" requireSignatureConfirmation="false"
    messageProtectionOrder="SignBeforeEncryptAndEncryptSignature"
    messageSecurityVersion="WSSecurity11WSTrustFebruary2005WSSecureConversationFebr
uary2005WSSecurityPolicy11BasicSecurityProfile10" />
    <textMessageEncoding messageVersion="Soap12" />
    <httpTransport authenticationScheme="Anonymous" />
</binding>
</customBinding>

```

The UsernameForCertificate policy requires that the client be configured with the service's X.509 certificate. This is accomplished using endpoint behavior:

```

<endpointBehaviors>
  <behavior name="ClientBehavior">
    <clientCredentials>
      <serviceCertificate>
        <defaultCertificate findValue="[certificate hash value]"
          storeLocation="LocalMachine" storeName="My"
          x509FindType="FindByThumbprint"/>
      </serviceCertificate>
    </clientCredentials>
  </behavior>
</endpointBehaviors>

```

The client must provide username/password credentials. This is accomplished with the proxy class:

```

EchoServiceClient client = new EchoServiceClient("Endpoint");
client.ClientCredentials.UserName.UserName = "Tester";
client.ClientCredentials.UserName.Password = "password";

```

## USERNAMEOVERTRANSPORT

The UsernameOverTransport policy is an Aggregate Policy that includes the following policies:

- UsernameOverTransport\_part1 (WS-Security Transport Binding Policy)
- UsernameOverTransport part2 (WS-Security Supporting Tokens Policy)

In this policy, the client authenticates with a Username Token which appears at the SOAP layer as a signed supporting token that is always sent from the initiator to the recipient. The service is authenticated using an X.509 certificate at the transport layer. The binding used is a transport binding.

### Supported by the Agent for WCF

Yes

### WCF Configuration

```

<customBinding>

```

```

<binding name="UserNameOverTransport">
  <security defaultAlgorithmSuite="Default"
authenticationMode="UserNameOverTransport"
  requireDerivedKeys="true" securityHeaderLayout="Lax" includeTimestamp="true"
  keyEntropyMode="CombinedEntropy"
  messageSecurityVersion="WSSecurity11WSTrustFebruary2005WSSecureConversationFebr
uary2005WSSecurityPolicy11BasicSecurityProfile10"/>
  <textMessageEncoding messageVersion="Soap12" />
  <httpsTransport authenticationScheme="Anonymous" />
</binding>
</customBinding>

```

The client must provide username/password credentials. This is accomplished with the proxy class:

```

EchoServiceClient client = new EchoServiceClient("Endpoint");
client.ClientCredentials.UserName.UserName = "Tester";
client.ClientCredentials.UserName.Password = "password";

```

## CERTIFICATEOVERTransport

The CertificateOverTransport policy is an Aggregate Policy that includes the following policies:

- CertificateOverTransport\_part1 (WS-Security Transport Binding Policy)
- CertificateOverTransport part2 (WS-Security Supporting Tokens Policy)

In this policy, the client authenticates using an X.509 certificate which appears at the SOAP layer as an endorsing supporting token that is always sent from the initiator to the recipient. The service is authenticated using an X.509 certificate at the transport layer. The binding used is a transport binding.

## Supported by Agent for WCF

Yes

## WCF Configuration

```

<customBinding>
  <binding name="CertificateOverTransport">
    <security defaultAlgorithmSuite="Default"
authenticationMode="CertificateOverTransport"
  requireDerivedKeys="true" securityHeaderLayout="Lax" includeTimestamp="true"
  keyEntropyMode="CombinedEntropy"
  messageSecurityVersion="WSSecurity11WSTrustFebruary2005WSSecureConversationFebr
uary2005WSSecurityPolicy11BasicSecurityProfile10"/>
    <textMessageEncoding messageVersion="Soap12" />
    <httpsTransport authenticationScheme="Anonymous" />
  </binding>
</customBinding>

```

The client must be configured with its own certificate. This is accomplished using endpoint behavior:

```

<endpointBehaviors>
  <behavior name="ClientBehavior">
    <clientCredentials>
      <clientCertificate findValue="9cbdd7b964f2113cc0182765c7293891b52591b4"
        storeLocation="LocalMachine" storeName="My"
        x509FindType="FindByThumbprint" />
    </clientCredentials>
  </behavior>
</endpointBehaviors>

```

## KERBEROSOVERTRANSPORT

The KerberosOverTransport policy is an Aggregate Policy that includes the following policies:

- KerberosOverTransport\_part1 (WS-Security Transport Binding Policy)
- KerberosOverTransport part2 (WS-Security Supporting Tokens Policy)

In this policy, the client authenticates to the service using a Kerberos ticket. The Kerberos token appears at the SOAP layer as an endorsing supporting token. The service is authenticated using an X.509 certificate at the transport layer. The binding is a transport binding.

### Supported by the Agent for WCF

Yes

### WCF Configuration

```

<customBinding>
  <binding name="KerberosOverTransport">
    <security defaultAlgorithmSuite="Basic128"
      authenticationMode="KerberosOverTransport"
      requireDerivedKeys="false" securityHeaderLayout="Lax" includeTimestamp="true"
      keyEntropyMode="CombinedEntropy"
      messageSecurityVersion="WSSecurity11WSTrustFebruary2005WSecureConversationFebr
        uary2005WSecurityPolicy11BasicSecurityProfile10"/>
    <textMessageEncoding messageVersion="Soap12" />
    <httpsTransport authenticationScheme="Anonymous" />
  </binding>
</customBinding>

```

The client has to provide Windows credentials that can be done either implicitly, using current process identity, or explicitly with the proxy class:

```

EchoServiceClient client = new EchoServiceClient("Endpoint");
client.ClientCredentials.Windows.ClientCredential.Domain = "DEV";
client.ClientCredentials.Windows.ClientCredential.UserName = "testuser";
client.ClientCredentials.Windows.ClientCredential.Password = "password";

```

## **SAMLOVERTRANSPORT**

The SAMLOverTransport policy is an Aggregate Policy that includes the following policies:

- SAMLOverTransport\_part1 (WS-Security Transport Binding Policy)
- SAMLOverTransport\_part2 (WS-Security Supporting Tokens Policy)

In this policy, the client authenticates with a SAML Token which appears at the SOAP layer as a signed supporting token that is always sent from the initiator to the recipient. The service is authenticated using an X.509 certificate at the transport layer. The binding used is a transport binding.

### **Supported by Agent for WCF**

No

## **CONFIGURING SOA SOFTWARE WS-AUDITING POLICIES**

Policy Manager provides the following policy types that are used to configure monitoring and recording on service operations.

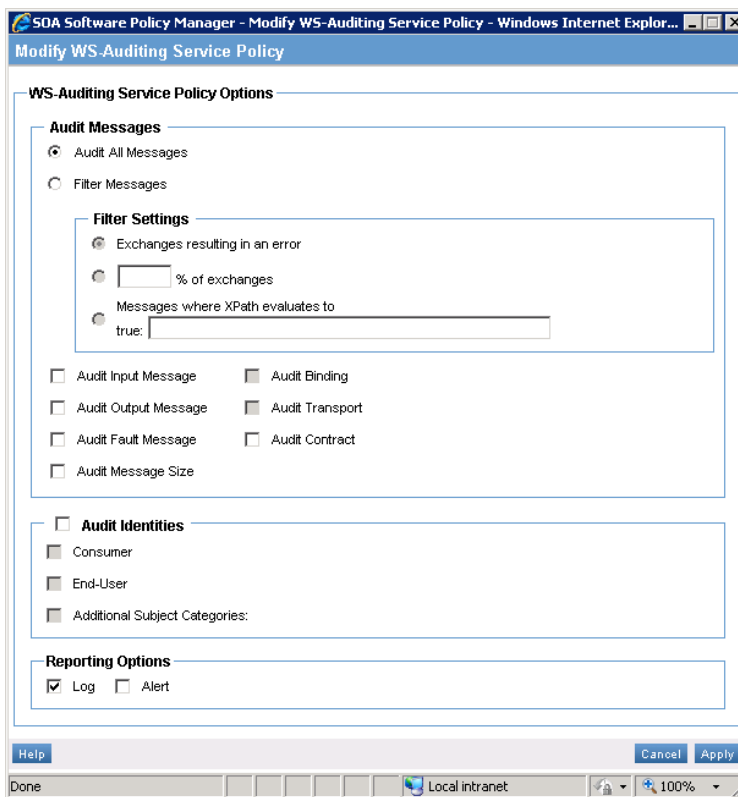
### **WS-Auditing Service Policy**

The WS-Auditing Service Policy can be assigned to a binding or a binding operation, and indicates when usage monitoring and recording will be applied to an intercepted message. The following policy assignment scenarios apply:

- If assigned to a binding, all messages in all operations will be monitored.
- If assigned to an operation, all messages related to the operation will be monitored.

This policy deals with an envelope-abstract message. This means if a SOAP message is intercepted, all provided XPath expressions will be evaluated starting from the first element inside the <soap:Body/>.





**Figure 2-1: Modify WS-Auditing Service Policy**

The Agent for WCF supports the following configuration options:

Name	Description
Audit All Messages	A checkbox that enables the auditing of all messages.
Filter Messages	<p>A checkbox that enables the ability to filter messages based on the following Filter Settings:</p> <ul style="list-style-type: none"> <li>• Exchanges resulting in an error—A radio button option that enables the ability to audit only request/response pairs that include fault messages.</li> <li>• Percentage of exchanges—A radio button option that enables the ability to audit message pairs as a random sample of messages based on a specified percentage. This option includes a text box that allows you to enter an integer percentage between 1 and 99.</li> <li>• Messages with XPath—A radio button option that enables the ability to audit messages that satisfy a specified XPath. The provided XPath is executed against request message content (starting from &lt;soap:Body/&gt;). If an expression evaluates to True, the message pair will be monitored.</li> </ul>
Audit Input Message	A checkbox that enables the recording of request messages. This option applies only to messages that passed the message filter defined in the "Filter Messages" section.

Name	Description
Audit Output Message	A checkbox that enables the recording of response messages. This option applies only to messages that passed the message filter defined in the "Filter Messages" section.
Audit Fault Message	A checkbox that enables the recording of fault messages.

The remaining policy options are not applicable to the Agent for WCF.

## WS-Auditing Message Policy

The WS-Auditing Message Policy can only be assigned to a message inside a binding and indicates what message content should be recorded. This policy can only be applied to a message belonging to an operation that has a WS-Auditing Service Policy assigned, or is inside a binding with such policy.

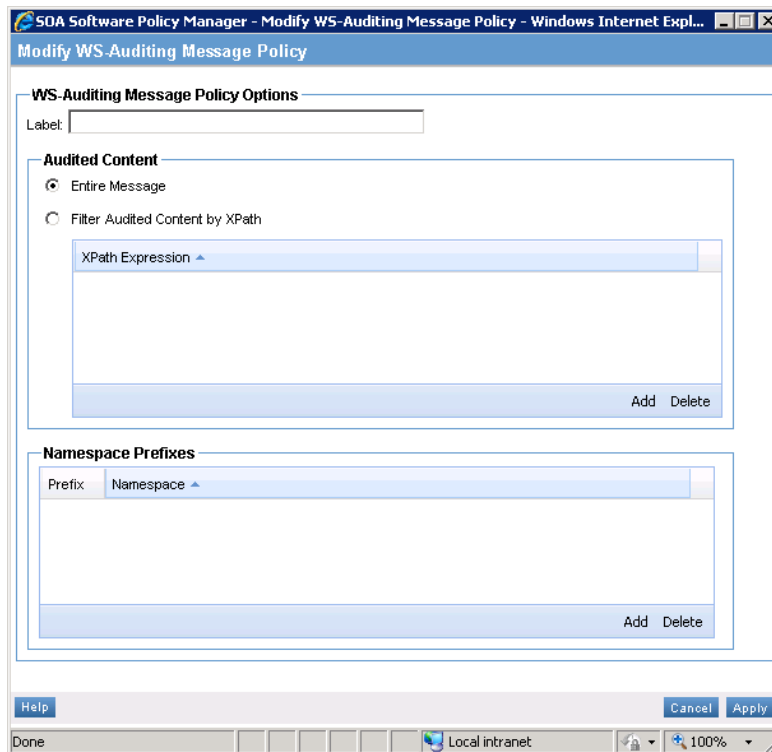


Figure 2-2: Modify WS-Auditing Message Policy

The Agent for WCF supports the following configuration options:

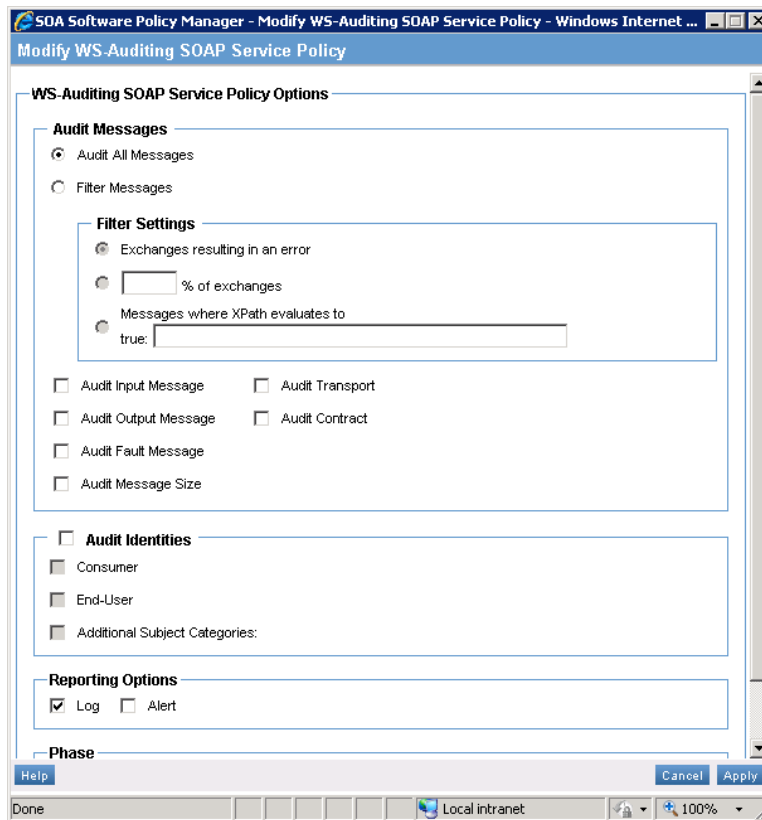
Name	Description
Label	A text box that allows you to enter the name of the recorded content in the audit record.
Entire Message	A checkbox that enables the auditing of all messages. This option audits all message exchanges (carried inside <soap:Body/> element).
Filter Audited Content by XPath	A checkbox that enables the ability to filter a subset of messages based on a defined filter (i.e., XPath expression). The "XPath Expression" table allows you to specify one or more XPath expressions that audited messages must satisfy. The table includes an "XPath Expression" table cell which is editable.
Namespace Prefixes	A table that allows you to specify an XPath that audited messages must satisfy. The "Namespace Prefixes" table allows you to enter one or more Namespace Prefix expressions for filtering messages. The table includes "Prefix" and "Namespace" table cells. The table cells are editable.

### WS-Auditing SOAP Service Policy

The WS-Auditing SOAP Service Policy is similar to the WS-Auditing Service Policy and can be assigned to a binding or a binding operation. It indicates when usage monitoring and recording will be applied to an intercepted message. The following policy assignment scenarios apply:

- If assigned to a binding, all messages in all operations will be monitored.
- If assigned to an operation, all messages related to the operation will be monitored.

In contrast with the WS-Auditing Service Policy, the WS-Auditing Service Policy works specifically with SOAP bindings so XPath expressions are evaluated starting from the <soap:Envelope/> element.



**Figure 2-3: Modify WS-Auditing SOAP Service Policy**

The set of options in the "WS-Auditing SOAP Service Policy" is similar to WS-Auditing Service Policy options with the addition of the Phase property. This option indicates where message recording should occur. If it is set to "Application," messages are recorded close to the service level (i.e., after the security header has already been processed and message content has been decoded). If "Phase" is set to "Wire," messages are recorded close to the transport (i.e., before message content has been modified by the channel stack). Note, that if recording is not enabled with the "Audit Input Message" or "Audit Output Message" option, this property has no effect on the message processing.

### WS-Auditing SOAP Message Policy

The WS-Auditing SOAP Message Policy is similar to the WS-Auditing Message Policy and can only be assigned to a message inside a binding. This policy indicates what SOAP message content should be recorded. In contrast with the WS-Auditing Message Policy, the WS-Auditing SOAP Message Policy works specifically with SOAP bindings and SOAP messages. XPath expressions are evaluated starting from <soap:Envelope/> element.

The Agent for WCF supports the following configuration options:

Name	Description
Audit SOAP Envelope Message Size	A checkbox that enables auditing of the SOAP envelope message size. If a message is recorded, then its size is also recorded.
Audit Transport Headers	<p>A checkbox that enables the ability to audit transport headers. If the message is recorded, then HTTP headers are always recorded for request messages.</p> <ul style="list-style-type: none"> <li>• All Headers—A radio button option that enables the ability to audit all transport headers.</li> <li>• Specific Headers—A radio button option that enables the ability to audit a list of transport headers. The "Transport Header" table stores a list transport header names to be audited. Transport headers can be added or deleted.</li> </ul> <p>The table includes a "Transport Header" table cell which is editable. One or more "Transport Headers" can be entered for filtering messages. The "Add" button adds an empty row to the "Transport Header" table. The "Remove" button deletes a row from the "Transport Header" table.</p>
Audit Wire Formatted Message Content	<p>A checkbox that enables auditing of message content in its wire format. Wire format is encoded for transport using a protocol. For incoming messages auditing occurs before SOAP headers are processed. For outgoing messages auditing occurs after all SOAP headers are applied. The following options are supported:</p> <ul style="list-style-type: none"> <li>• Entire Message—A checkbox that enables the auditing of all messages.</li> <li>• Filter Audited Content by XPath—A checkbox that enables the ability to filter a subset of messages based on a defined filter (i.e., XPath expression).</li> </ul> <p>The "XPath Expression" table allows you to specify one or more XPath expressions that audited messages must satisfy. The table includes an "XPath Expression" table cell which is editable.</p> <p>XPath Expression—A table row that allows you to specify the "XPath Expression" to be used for filtering. The "Add" button adds an empty "XPath Expression" row to the "XPath Expression" table. The "Remove" button deletes a row from the "XPath Expression" table.</p>
Audit Application Formatted Message Content	<p>A checkbox that enables auditing of message content in its application readable format. This indicates that the message must be recorded close to the service after the security header has already been processed and message content has been decoded. The application is independent of protocol in the form the application consumes.</p> <p>For incoming messages auditing occurs after all SOAP headers have been processed. In this scenario, the SOAP envelope may remain. For</p>

Name	Description
	<p>outgoing messages auditing occurs before SOAP headers have been added. Some headers may be present if included by the application, or service, itself.</p> <p>The following options are supported:</p> <ul style="list-style-type: none"> <li>• Entire Message—A checkbox that enables the auditing of all messages.</li> <li>• Filter Audited Content by XPath—A checkbox that enables the ability to filter messages based on XPath expressions.</li> </ul> <p>An "XPath Expression" table allows you to specify an XPath that audited messages must satisfy. The table includes an "XPath Expression" table cell which is editable. One or more XPath Expressions can be entered for filtering messages.</p> <p>XPath Expression—A table row that allows you to specify the "XPath Expression" to be used for filtering. The "Add" button adds an empty "XPath Expression" row to the "XPath Expression" table. The "Remove" button deletes a row from the "XPath Expression" table.</p>
Namespace Prefixes	<p>A table that allows you to map XML namespaces to prefixes that can be used in XPath expressions that audited messages must satisfy.</p> <p>The "Namespace Prefixes" table allows you to enter one or more Namespace Prefix expressions for filtering messages. The table includes "Prefix" and "Namespace" table cells. The table cells are editable.</p> <ul style="list-style-type: none"> <li>• Prefix—A table row that allows you to specify the "Prefix" of the "Namespace" used in the XPath.</li> <li>• Namespace—A table row that allows you to specify the "Namespace" associated with the "Prefix."</li> </ul> <p>The "Add" button adds an empty "Prefix" and "Namespace" row to the "Namespace Prefixes" table. The "Remove" button deletes a row from the "Namespace Prefixes" table.</p>

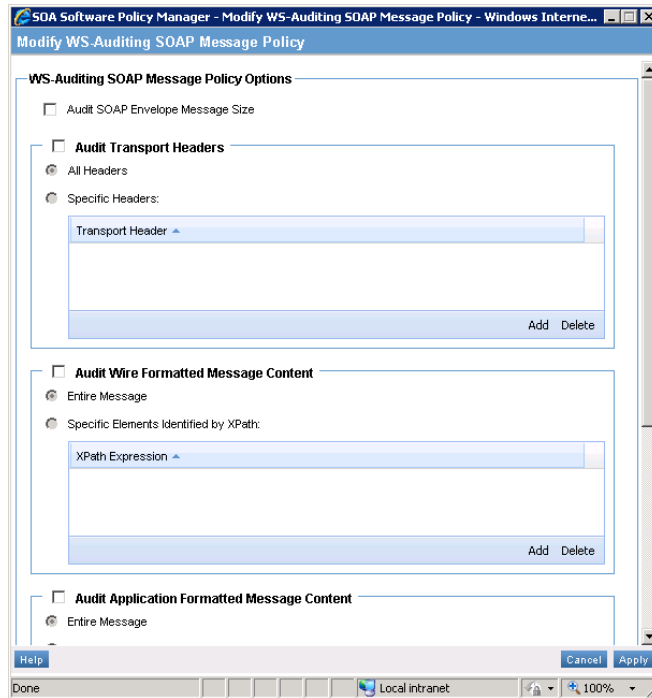


Figure 2-4: Modify WS-Auditing SOAP Message Policy #1

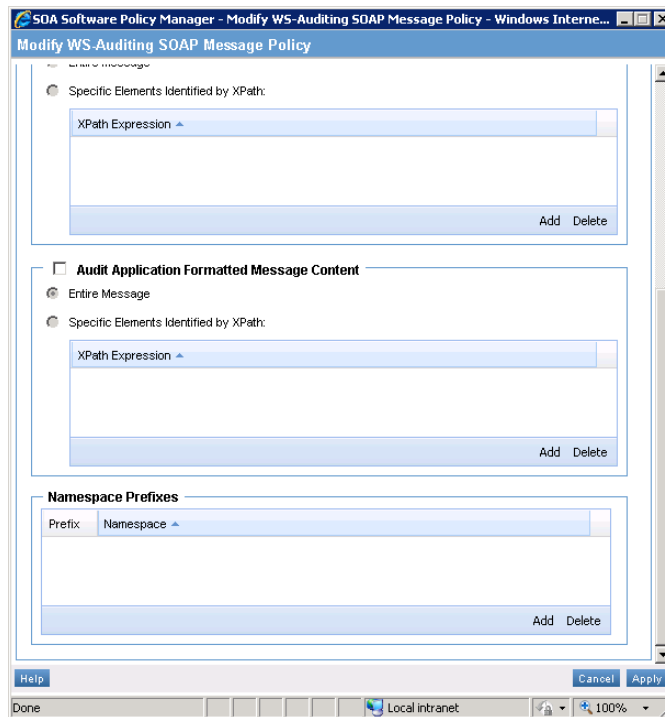


Figure 2-5: Modify WS-Auditing SOAP Message Policy #2

## USAGE DATA MONITORING

Policy Manager includes two sample auditing policies that can be assigned to service bindings or operations inside a binding. Both policies are based on *WS-Auditing Service Policy* types, which deal with the message content inside SOAP body.

Name	Description
BasicAuditing	Provides basic auditing of messages. Message metrics are recorded and viewable via the Monitoring > Logs tab. The messages themselves are not recorded.
DetailedAuditing	Provides detailed auditing of messages. Message metrics are recorded and viewable via the Monitoring > Logs tab. With this policy, the entire message of each exchange is recorded.



# Chapter 3: Policy Manager Configurations for WCF Common Security Scenarios

## OVERVIEW

Microsoft WCF Common Security Scenarios (<http://msdn.microsoft.com/en-us/library/ms730301.aspx>) define typical Intranet and Internet scenarios that are described by the variations of the Microsoft WCF wsHttpBinding and netTcpBinding configurations.

This chapter describes interoperable security and Microsoft-specific use cases.

## CONFIGURE SERVICES

To configure services registered with Policy Manager with Microsoft WCF Common Security Scenarios, begin by registering physical services in Policy Manager. Services should not have policies attached. Each service must include a SOAP 1.2 binding. A WCF service endpoint for the configuration may be described by the following customBinding:

```
<customBinding>
  <binding name="BasicHttpBindingSoap12">
    <textMessageEncoding messageVersion="Soap12"/>
    <httpTransport/>
  </binding>
</customBinding>
```

If you start with registering a physical service with a SOAP 1.1 endpoint binding instead (e.g., endpoint is configured with the default WCF basicHttpBinding), the WCF configurations will still be valid after applying the procedures described below. The effective WCF configurations will then be equivalent to variations of the WCF customBinding rather than wsHttpBinding or netTcpBinding.

Alternatively, you can create a new SOAP 1.2 binding for an existing service using the "Add Binding" function available in the "Configure > Registry > Bindings" section of the Policy Manager "Management Console."

### To Configure a Binding

Step	Procedure
1.	In the Organization Tree of the Policy Manager "Management Console" select the Configure > Registry > Bindings > Tab. The "Bindings Summary" displays.
2.	Click "Add Binding." The "Add Binding Wizard" launches and the "Select Interfaces" screen displays.
3.	Select an interface and click <b>Next</b> to continue. The "Specify Binding Details" screen displays.
4.	Enter the "Namespace URI" and "Localpart" elements and optional "Description" to your binding definition. Select "SOAP 1.2" from the "Binding Type" drop-down list box. Click <b>Next</b> to continue. The "SOAP 1.2 Binding Details" screen displays.
5.	Configure the SOAP 1.2 binding properties and click <b>Finish</b> to save your binding.
6.	After saving the binding, navigate to the "Service Details" screen of the service you would like to add the binding to.
7.	In the "Services Overview" portlet, navigate to the "Interfaces and Bindings" section and click "Manage." The "Manage Interfaces and Bindings Wizard" launches.
8.	On the "Select Interfaces" screen search for and select the interface of the binding you created using the "Add Binding Wizard" and copy it to the "Interfaces Assigned" panel. Click <b>Next</b> to continue. The "Select Binding" screen displays.
9.	On the "Select Binding" screen, click the checkbox of the binding name that was created using the "Add Binding Wizard."
10.	Click "Finish" to add the binding to the current service.
11.	To delete an existing binding, uncheck the binding line item on the "Select Binding" screen.
12.	Refer to the "Policy Manager Online Help" for more information on the specified Policy Manager functions.

## TRANSPORT SECURITY WITH BASIC AUTHENTICATION

The following illustration shows a Windows Communication Foundation (WCF) service and client. The server needs a valid X.509 certificate that can be used for Secure Sockets Layer (SSL), and the clients must trust the server's certificate (<http://msdn.microsoft.com/en-us/library/ms733775.aspx>).

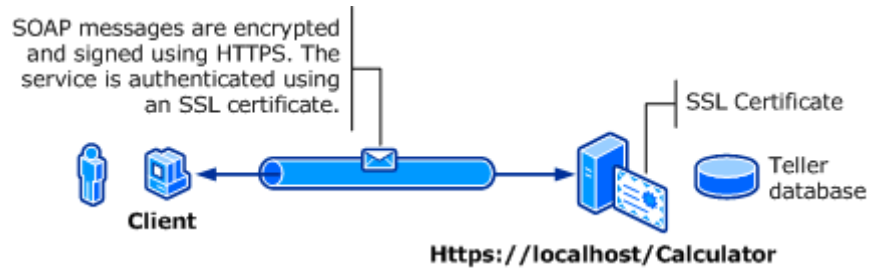


Figure 3-1: WCF Service and Client

## Interoperability

Yes

## WCF Configuration

```
<wsHttpBinding>
  <binding name="BasicAuthOverHttps">
    <security mode="Transport">
      <transport clientCredentialType="Basic" />
    </security>
  </binding>
</wsHttpBinding>
```

## Policy Manager Configuration

The following procedure illustrates how to configure Transport Security with Basic Authentication.

### To Configure Transport Security with Basic Authentication

Step	Procedure
1.	In the Organization Tree of the Policy Manager "Management Console" select the Organization > Policies Folder > Operational Tab.
2.	Click "Add Policy." The "Add Policy Wizard" launches and the "Select Policy Creation Option" screen displays.
3.	Click the "Add Policy" radio button. From the "Type" drop-down list box select "XML Policy." Click <b>Next</b> . On the "Specify Policy Details" screen enter the policy "Name" (e.g., WCF Basic Authentication) and optional "Description."
4.	To save the policy click <b>Finish</b> then <b>Close</b> . The "Policy Details" screen for the XML Policy displays.
5.	In the "XML Policy" portlet, click "Modify." The "Modify XML Policy" screen displays. In the "XML Policy Content" text box insert the following XML Policy definition and click <b>Apply</b> to save the policy definition: <pre>&lt;wsp:Policy wsu:Id="WSHttpBinding_BasicAuth"   xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"   xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"   xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"</pre>

## To Configure Transport Security with Basic Authentication

	<pre>       &gt;       &lt;wsp:ExactlyOne&gt;         &lt;wsp:All&gt;           &lt;http:BasicAuthentication xmlns:http="http://schemas.microsoft.com/ws/06/2004/policy/http"/&gt;           &lt;sp:TransportBinding xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy"&gt;             &lt;wsp:Policy&gt;               &lt;sp:TransportToken&gt;                 &lt;wsp:Policy&gt;                   &lt;sp:HttpsToken RequireClientCertificate="false"/&gt;                 &lt;/wsp:Policy&gt;               &lt;/sp:TransportToken&gt;               &lt;sp:AlgorithmSuite&gt;                 &lt;wsp:Policy&gt;                   &lt;sp:Basic256/&gt;                 &lt;/wsp:Policy&gt;               &lt;/sp:AlgorithmSuite&gt;               &lt;sp:Layout&gt;                 &lt;wsp:Policy&gt;                   &lt;sp:Strict/&gt;                 &lt;/wsp:Policy&gt;               &lt;/sp:Layout&gt;             &lt;/wsp:Policy&gt;           &lt;/sp:TransportBinding&gt;           &lt;wsaw:UsingAddressing/&gt;         &lt;/wsp:All&gt;       &lt;/wsp:ExactlyOne&gt;     &lt;/wsp:Policy&gt; </pre>
--	--

The following procedure illustrates how to attach a policy to a service endpoint.

## To Attach Policy to Service Endpoint

Step	Procedure
1.	To attach a new policy to a service endpoint, in the Organization Tree of the Policy Manager "Management Console" select the service you would like to attach a policy to. The "Service Details" screen displays.
2.	Select the "Bindings" tab. Click on the binding "Qualified Name." The "Binding Details" screen displays. In the "Operational Policy Attachments" portlet click "Manage." The "Manage Operational Policy Attachments" screen displays.
3.	Navigate the tree hierarchy and click the checkbox of one or more policies you would like to attach to the current binding. In this example, select the XML Policy you just created. Click <b>Apply</b> . The selected policies are added to the "Manage Operational Policy Attachments" screen.
4.	Confirm that the service Access Point address is defined with the HTTPS Transport Schema. If it is not, click the "Access Points" tab of the current service. On the Access Point line item select "Modify Access Point" from the "Actions" drop-down list box. On the "Specify Access Point Details" screen click <b>Next</b> . The "Specify Details" screen displays for the Access Point binding. Modify the endpoint address and click <b>Finish</b> .

## Testing Service

If your physical service is managed by the Agent for WCF and it is tested by the WCF client application managed by the SOA Software WCF Delegate, reconfiguring with either a service or client application is not required as they are both managed by SOA Software. Note: if a service is deployed in IIS Server, confirm that Basic Authentication is enabled for the virtual directory that hosts your service. If Basic Authentication is not enabled, the WCF service will throw an exception at startup.

If you are testing a service with an external tool that is not managed by the SOA Software Delegate, use the service metadata URL ("WSDL URL" link) on the "Service Overview" portlet to generate a proxy, sample messages, and configurations relevant to your external tool.

## MESSAGE SECURITY WITH A WINDOWS CLIENT OVER HTTP

This scenario is the default configuration of the wsHttpBinding and uses message level security based on SSPI Negotiation (NTLM/Kerberos) and Secure Conversation protocols.

### Interoperability

Yes

### WCF Configuration

```
<wsHttpBinding>
  <binding name="Default" />
</wsHttpBinding>
```

### Policy Manager Configuration

The following procedure illustrates how to configure Message Security with a Windows Client over HTTP.

#### To Configure Message Security with a Windows Client over HTTP

Step	Procedure
1.	In the Organization Tree of the Policy Manager "Management Console" select Organization > Policies Folder > Operational Tab.
2.	Click "Add Policy." The "Add Policy Wizard" launches and the "Select Policy Creation Option" screen displays.
3.	Click the "Add Policy" radio button. From the "Type" drop-down list box select "Aggregate Policy." Click <b>Next</b> . On the "Specify Policy Details" screen enter the policy "Name" (e.g., WSHttp-Default) and optional "Description."
4.	To save the policy click <b>Finish</b> then <b>Close</b> . The "Policy Details" screen for the

**To Configure Message Security with a Windows Client over HTTP**

	Aggregate Policy displays.
5.	In the "Aggregate Policy" portlet, click <b>Add</b> . The "Add Policy Wizard" launches and the "Select Policy Creation Option" screen displays.
6.	Click the "Add Policy" radio button. From the "Type" drop-down list box select "XML Policy." Click <b>Next</b> . On the "Specify Policy Details" screen enter the policy "Name" (e.g., Binding Policy) and optional "Description."
7.	To save the policy click <b>Finish</b> then <b>Close</b> . The "Policy Details" screen for the "XML Policy" displays.
8.	<p>In the "XML Policy" portlet, click <b>Modify</b>. The "Modify XML Policy" screen displays. In the "XML Policy Content" text box insert the following XML Policy definition and click <b>Apply</b> to save the policy definition:</p> <pre> &lt;wsp:Policy wsu:Id="WSHttpBinding_Default"   xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"   xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"   xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"   &gt;   &lt;wsp:ExactlyOne&gt;     &lt;wsp&gt;All&gt;       &lt;sp:SymmetricBinding xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy"&gt;         &lt;wsp:Policy&gt;           &lt;sp:ProtectionToken&gt;             &lt;wsp:Policy&gt;               &lt;sp:SecureConversationToken sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/IncludeToken/AlwaysToRecipient"&gt;                 &lt;wsp:Policy&gt;                   &lt;sp:RequireDerivedKeys/&gt;                   &lt;sp:BootstrapPolicy&gt;                     &lt;wsp:Policy&gt;                       &lt;sp:SignedParts&gt;                         &lt;sp:Body/&gt;                         &lt;sp:Header Name="To" Namespace="http://www.w3.org/2005/08/addressing"/&gt;                         &lt;sp:Header Name="From" Namespace="http://www.w3.org/2005/08/addressing"/&gt;                         &lt;sp:Header Name="FaultTo" Namespace="http://www.w3.org/2005/08/addressing"/&gt;                         &lt;sp:Header Name="ReplyTo" Namespace="http://www.w3.org/2005/08/addressing"/&gt;                         &lt;sp:Header Name="MessageID" Namespace="http://www.w3.org/2005/08/addressing"/&gt;                         &lt;sp:Header Name="RelatesTo" Namespace="http://www.w3.org/2005/08/addressing"/&gt;                         &lt;sp:Header Name="Action" Namespace="http://www.w3.org/2005/08/addressing"/&gt;                       &lt;/sp:SignedParts&gt;                       &lt;sp:EncryptedParts&gt;                         &lt;sp:Body/&gt;                       &lt;/sp:EncryptedParts&gt;                     &lt;sp:SymmetricBinding&gt;                       &lt;wsp:Policy&gt;                         &lt;sp:ProtectionToken&gt;                           &lt;wsp:Policy&gt; </pre>


## To Configure Message Security with a Windows Client over HTTP

```

        <sp:SpnegoContextToken
sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/IncludeToken/AlwaysToRecipient" >
            <wsp:Policy>
                <sp:RequireDerivedKeys/>
            </wsp:Policy>
        </sp:SpnegoContextToken>
    </wsp:Policy>
</sp:ProtectionToken>
<sp:AlgorithmSuite>
    <wsp:Policy>
        <sp:Basic256/>
    </wsp:Policy>
</sp:AlgorithmSuite>
<sp:Layout>
    <wsp:Policy>
        <sp:Strict/>
    </wsp:Policy>
</sp:Layout>
<sp:IncludeTimestamp/>
<sp:EncryptSignature/>
<sp:OnlySignEntireHeadersAndBody/>
</wsp:Policy>
</sp:SymmetricBinding>
<sp:Wss11>
    <wsp:Policy>
        <sp:MustSupportRefKeyIdentifier/>
        <sp:MustSupportRefIssuerSerial/>
        <sp:MustSupportRefThumbprint/>
        <sp:MustSupportRefEncryptedKey/>
    </wsp:Policy>
</sp:Wss11>
<sp:Trust10>
    <wsp:Policy>
        <sp:MustSupportIssuedTokens/>
        <sp:RequireClientEntropy/>
        <sp:RequireServerEntropy/>
    </wsp:Policy>
</sp:Trust10>
</wsp:Policy>
</sp:BootstrapPolicy>
</wsp:Policy>
</sp:SecureConversationToken>
</wsp:Policy>
</sp:ProtectionToken>
<sp:AlgorithmSuite>
    <wsp:Policy>
        <sp:Basic256/>
    </wsp:Policy>
</sp:AlgorithmSuite>
<sp:Layout>
    <wsp:Policy>
        <sp:Strict/>
    </wsp:Policy>
</sp:Layout>
<sp:IncludeTimestamp/>
<sp:EncryptSignature/>
<sp:OnlySignEntireHeadersAndBody/>
</wsp:Policy>
</sp:SymmetricBinding>
<sp:Wss11

```

### To Configure Message Security with a Windows Client over HTTP

	<pre> xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy"&gt;   &lt;wsp:Policy&gt;     &lt;sp:MustSupportRefKeyIdentifier/&gt;     &lt;sp:MustSupportRefIssuerSerial/&gt;     &lt;sp:MustSupportRefThumbprint/&gt;     &lt;sp:MustSupportRefEncryptedKey/&gt;   &lt;/wsp:Policy&gt; &lt;/sp:Wss11&gt; &lt;sp:Trust10 xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy"&gt;   &lt;wsp:Policy&gt;     &lt;sp:MustSupportIssuedTokens/&gt;     &lt;sp:RequireClientEntropy/&gt;     &lt;sp:RequireServerEntropy/&gt;   &lt;/wsp:Policy&gt; &lt;/sp:Trust10&gt; &lt;wsaw:UsingAddressing/&gt; &lt;/wsp:All&gt; &lt;/wsp:ExactlyOne&gt; &lt;/wsp:Policy&gt; </pre>
9.	<p>After the XML Policy is saved, select the WsHttp-Default "Aggregate Policy" in the Policies &gt; Organizational Polices folder of the Organization Tree. The "Aggregate Policy Details" screen displays.</p>
10.	<p>In the "Aggregate Policy" portlet, click "Add." The "Add Policy Wizard" launches and the "Select Policy Creation Option" screen displays.</p>
11.	<p>Click the "Add Policy" radio button. From the "Type" drop-down list box select "WS-Security Message Policy." Click <b>Next</b>. On the "Specify Policy Details" screen enter the policy "Name" (e.g., Message Policy) and optional "Description."</p>
12.	<p>To save the policy click <b>Finish</b> then <b>Close</b>. The "Policy Details" screen for the "XML Policy" displays.</p>
13.	<p>After these steps are complete the WsHttp-Default "Aggregate Policy" will include the two policies shown below:</p> <div style="text-align: center;">  </div> <p style="text-align: center;"><b>Figure 3-2: WsHttp-Default Aggregate Policy—via Organization Tree</b></p>

The following procedure illustrates how to attach a policy to a service endpoint.

### To Attach Policy to Service Endpoint

Step	Procedure
1.	<p>To attach a new policy to a service endpoint, in the Organization Tree of the Policy Manager "Management Console" select the service you would like to attach a policy to. The "Service Details" screen displays.</p>
2.	<p>Select the "Bindings" tab. Click on the binding "Qualified Name." The "Binding Details"</p>



**To Attach Policy to Service Endpoint**

	screen displays. In the "Operational Policy Attachments" portlet click "Manage." The "Manage Operational Policy Attachments" screen displays.
3.	Navigate the tree hierarchy, and click the checkbox of one or more policies you would like to attach to the current binding. In this example, select the Aggregate Policy you just created. Click <b>Apply</b> . The selected policies are added to the "Manage Operational Policy Attachments" screen.

**Testing Service**

If your physical service is managed by the Agent for WCF and it is tested by the WCF Client m2application managed by the SOA Software WCF Delegate, reconfiguring with either a service or client application is not required as they are both managed by SOA Software. Note: if a service is deployed in IIS Server, confirm that Basic Authentication is enabled for the virtual directory that hosts your service. If Basic Authentication is not enabled, the WCF service will throw an exception at startup.

If you are testing a service with an external tool that is not managed by the SOA Software Delegate, use the service metadata URL ("WSDL URL" link) on the "Service Overview" portlet to generate a proxy, sample messages, and configurations relevant to your external tool.

**MESSAGE SECURITY WITH A WINDOWS CLIENT OVER NET.TCP**

This scenario is the default configuration of the netTcpBinding and uses transport level security based on Windows Stream Security (NTLM/Kerberos) protocol.

**Interoperability**

Windows Communication Foundation Only

**WCF Configuration**

```
<netTcpBinding>
  <binding name="Default" />
</netTcpBinding>
```

**Policy Manager Configuration**

The following procedure illustrates how to configure Message Security with a Windows Client over NET.TCP.

### To Configure Message Security with a Windows Client over NET.TCP

Step	Procedure
1.	In the Organization Tree of the Policy Manager "Management Console" select Organization > Policies Folder > Operational Tab.
2.	Click "Add Policy." The "Add Policy Wizard" launches and the "Select Policy Creation Option" screen displays.
3.	Click the "Add Policy" radio button. From the "Type" drop-down list box select "Aggregate Policy." Click <b>Next</b> . On the "Specify Policy Details" screen enter the policy "Name" (e.g., NetTcp-Default) and optional "Description."
4.	To save the policy click <b>Finish</b> then <b>Close</b> . The "Policy Details" screen for the Aggregate Policy displays.
5.	In the "Aggregate Policy" portlet, click "Add." The "Add Policy Wizard" launches and the "Select Policy Creation Option" screen displays. Click the "Add Policy" radio button. From the "Type" drop-down list box select "XML Policy." Click <b>Next</b> . On the "Specify Policy Details" screen enter the policy "Name" (e.g., Binding Policy) and optional "Description. To save the policy click <b>Finish</b> then <b>Close</b> . The "Policy Details" screen for the "XML Policy" displays.
6.	<p>In the "XML Policy" portlet, click "Modify." The "Modify XML Policy" screen displays. In the "XML Policy Content" text box insert the following XML Policy definition and click <b>Apply</b> to save the policy definition:</p> <pre> &lt;wsp:Policy wsu:Id="NetTcpBinding_Default"   xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"   xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"   xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"   &gt;   &lt;wsp:ExactlyOne&gt;     &lt;wsp&gt;All&gt;       &lt;msb:BinaryEncoding xmlns:msb="http://schemas.microsoft.com/ws/06/2004/mspolicy/netbinary1"/&gt;       &lt;sp:TransportBinding xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy"&gt;         &lt;wsp:Policy&gt;           &lt;sp:TransportToken&gt;             &lt;wsp:Policy&gt;               &lt;msf:WindowsTransportSecurity xmlns:msf="http://schemas.microsoft.com/ws/2006/05/framing/policy"&gt;                 &lt;msf:ProtectionLevel&gt;EncryptAndSign&lt;/msf:ProtectionLevel&gt;               &lt;/msf:WindowsTransportSecurity&gt;             &lt;/wsp:Policy&gt;           &lt;/sp:TransportToken&gt;           &lt;sp:AlgorithmSuite&gt;             &lt;wsp:Policy&gt;               &lt;sp:Basic256/&gt;             &lt;/wsp:Policy&gt;           &lt;/sp:AlgorithmSuite&gt;           &lt;sp:Layout&gt;             &lt;wsp:Policy&gt;               &lt;sp:Strict/&gt;             &lt;/wsp:Policy&gt;           &lt;/sp:Layout&gt;         &lt;/wsp:Policy&gt;       &lt;/sp:TransportBinding&gt;     &lt;/wsp&gt;All&gt;   &lt;/wsp:ExactlyOne&gt; &lt;/wsp:Policy&gt; </pre>

### To Configure Message Security with a Windows Client over NET.TCP

	<pre> &lt;/sp:Layout&gt; &lt;/wsp:Policy&gt; &lt;/sp:TransportBinding&gt; &lt;wsaw:UsingAddressing/&gt; &lt;/wsp:All&gt; &lt;/wsp:ExactlyOne&gt; &lt;/wsp:Policy&gt; </pre>
--	---

The following procedure illustrates how to attach a policy to a service endpoint.

### To Attach Policy to Service Endpoint

Step	Procedure
1.	To attach a new policy to a service endpoint, in the Organization Tree of the Policy Manager "Management Console" select the service you would like to attach a policy to. The "Service Details" screen displays.
2.	Select the "Bindings" tab. Click on the binding "Qualified Name." The "Binding Details" screen displays. In the "Operational Policy Attachments" portlet click "Manage." The "Manage Operational Policy Attachments" screen displays.
3.	Navigate the tree hierarchy, and click the checkbox of one or more policies you would like to attach to the current binding. In this example, select the XML policy you just created. Click <b>Apply</b> . The selected policies are added to the "Manage Operational Policy Attachments" screen.
4.	Note: Microsoft WCF netTcpBinding and the NetTcp-Default policy (defined above) require an endpoint with a net.tcp transport address rather than HTTP or HTTPS address. When the NetTcp-Default policy is attached to a service endpoint binding, the endpoint address must conform to the net.tcp:// transport schema, otherwise create a new binding with net.tcp schema ( <code>transport=<a href="http://schemas.microsoft.com/soap/tcp">http://schemas.microsoft.com/soap/tcp</a></code> ).

### Testing Service

If your physical service is managed by the Agent for WCF and it is tested by the WCF client application managed by the SOA Software WCF Delegate, reconfiguring with either a service or the client application is not required as they are both managed by SOA Software. Note: if a service is deployed in IIS Server, confirm that Basic Authentication is enabled for the virtual directory that hosts your service. If Basic Authentication is not enabled, the WCF service will throw an exception at startup.

If you are testing a service with an external tool that is not managed by the SOA Software Delegate, use the service metadata URL ("WSDL URL" link) on the "Service Overview" portlet to generate a proxy, sample messages, and configurations relevant to your external tool.

## MESSAGE SECURITY WITH A USER NAME CLIENT

The following illustration shows a Windows Communication Foundation (WCF) service and client secured using message-level security. The service is authenticated with an X.509 certificate. The client authenticates using a user name and password (<http://msdn.microsoft.com/en-us/library/ms731058.aspx>).

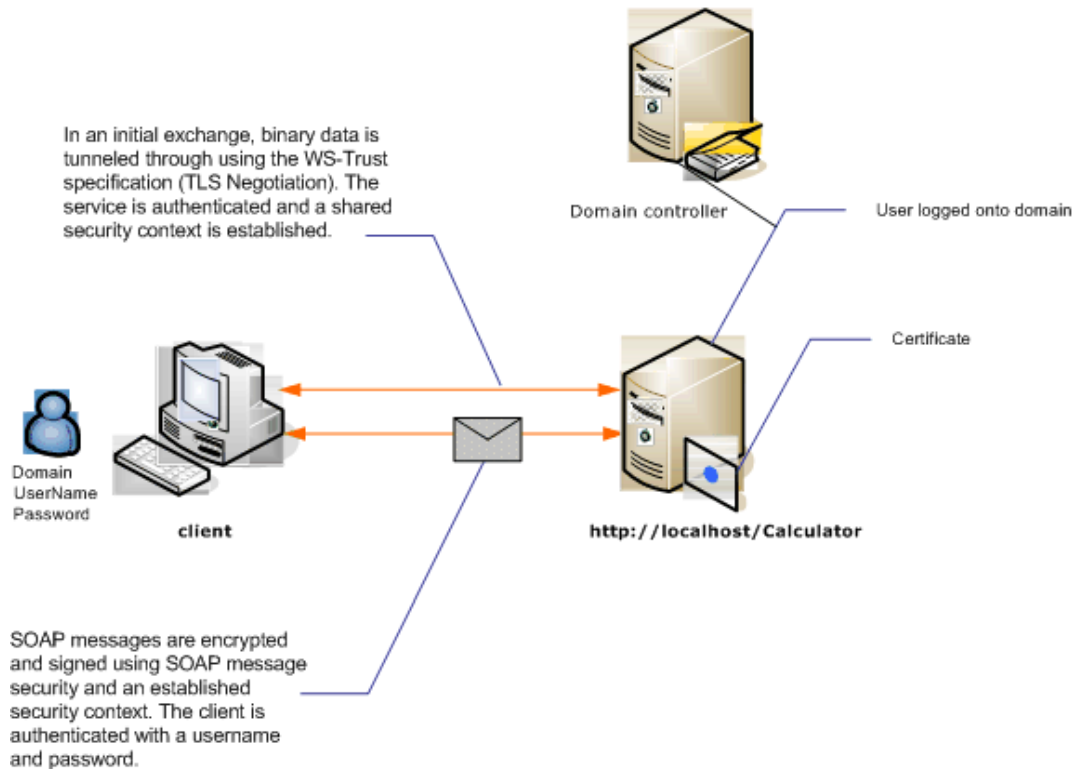


Figure 3-3: (WCF) Service and Client Secured using Message-level Security

## Interoperability

Windows Communication Foundation Only

## WCF Configuration

```
<wsHttpBinding>
  <binding name="MessageSecurityWithUsernameClient">
    <security mode="Message">
      <message clientCredentialType="UserName"/>
    </security>
  </binding>
</wsHttpBinding>
```

## Policy Manager Configuration

The following procedure illustrates how to configure Message Security with a User Name Client.

### To Configure Message Security with a User Name Client

Step	Procedure
1.	In the Organization Tree of the Policy Manager "Management Console" select Organization > Policies Folder > Operational Tab.
2.	Click "Add Policy." The "Add Policy Wizard" launches and the "Select Policy Creation Option" screen displays.
3.	Click the "Add Policy" radio button. From the "Type" drop-down list box select "Aggregate Policy." Click <b>Next</b> . On the "Specify Policy Details" screen enter the policy "Name" (e.g., MessageSecurityWithUsernameClient) and optional "Description."
4.	To save the policy click <b>Finish</b> then <b>Close</b> . The "Policy Details" screen for the Aggregate Policy displays.
5.	In the "Aggregate Policy" portlet, click "Add." The "Add Policy Wizard" launches and the "Select Policy Creation Option" screen displays. Click the "Add Policy" radio button. From the "Type" drop-down list box select "XML Policy." Click <b>Next</b> .
6.	On the "Specify Policy Details" screen enter the policy "Name" (e.g., MessageSecurityWithUsernameClient-Binding) and optional "Description." To save the policy click <b>Finish</b> then <b>Close</b> . The "Policy Details" screen for the "XML Policy" displays.
7.	In the "XML Policy" portlet, click "Modify." The "Modify XML Policy" screen displays. In the "XML Policy Content" text box insert the following XML Policy definition and click <b>Apply</b> to save the policy definition: <pre> &lt;wsp:Policy wsu:Id="WSHttpBinding_MessageSecurityWithUsernameClient"   xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"   xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"   xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"   &gt;   &lt;wsp:ExactlyOne&gt;     &lt;wsp:All&gt;       &lt;sp:SymmetricBinding xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy"&gt;         &lt;wsp:Policy&gt;           &lt;sp:ProtectionToken&gt;             &lt;wsp:Policy&gt;               &lt;sp:SecureConversationToken sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/IncludeToken/AlwaysToRecipient"&gt;                 &lt;wsp:Policy&gt;                   &lt;sp:RequireDerivedKeys/&gt;                   &lt;sp:BootstrapPolicy&gt;                     &lt;wsp:Policy&gt;                       &lt;sp:SignedParts&gt;                         &lt;sp:Body/&gt; </pre>

## To Configure Message Security with a User Name Client

```

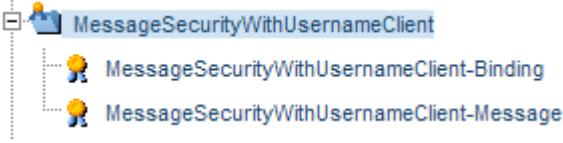
        <sp:Header Name="To"
Namespace="http://www.w3.org/2005/08/addressing" />
        <sp:Header Name="From"
Namespace="http://www.w3.org/2005/08/addressing" />
        <sp:Header Name="FaultTo"
Namespace="http://www.w3.org/2005/08/addressing" />
        <sp:Header Name="ReplyTo"
Namespace="http://www.w3.org/2005/08/addressing" />
        <sp:Header Name="MessageID"
Namespace="http://www.w3.org/2005/08/addressing" />
        <sp:Header Name="RelatesTo"
Namespace="http://www.w3.org/2005/08/addressing" />
        <sp:Header Name="Action"
Namespace="http://www.w3.org/2005/08/addressing" />
    </sp:SignedParts>
    <sp:EncryptedParts>
    <sp:Body/>
    </sp:EncryptedParts>
    <sp:SymmetricBinding>
    <wsp:Policy>
    <sp:ProtectionToken>
    <wsp:Policy>
    <mssp:SslContextToken
sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/Inc
ludeToken/AlwaysToRecipient"
xmlns:mssp="http://schemas.microsoft.com/ws/2005/07/securitypolicy">
    <wsp:Policy>
    <sp:RequireDerivedKeys/>
    </wsp:Policy>
    </mssp:SslContextToken>
    </wsp:Policy>
    </sp:ProtectionToken>
    <sp:AlgorithmSuite>
    <wsp:Policy>
    <sp:Basic256/>
    </wsp:Policy>
    </sp:AlgorithmSuite>
    <sp:Layout>
    <wsp:Policy>
    <sp:Strict/>
    </wsp:Policy>
    </sp:Layout>
    <sp:IncludeTimestamp/>
    <sp:EncryptSignature/>
    <sp:OnlySignEntireHeadersAndBody/>
    </wsp:Policy>
    </sp:SymmetricBinding>
    <sp:SignedSupportingTokens>
    <wsp:Policy>
    <sp:UsernameToken
sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/Inc
ludeToken/AlwaysToRecipient">
    <wsp:Policy>
    <sp:WssUsernameToken10/>
    </wsp:Policy>
    </sp:UsernameToken>
    </wsp:Policy>
    </sp:SignedSupportingTokens>
    <sp:Wss11>
    <wsp:Policy>
    <sp:MustSupportRefKeyIdentifier/>

```

## To Configure Message Security with a User Name Client

	<pre>         &lt;sp:MustSupportRefIssuerSerial/&gt;         &lt;sp:MustSupportRefThumbprint/&gt;         &lt;sp:MustSupportRefEncryptedKey/&gt;       &lt;/wsp:Policy&gt;     &lt;/sp:Wss11&gt;     &lt;sp:Trust10&gt;       &lt;wsp:Policy&gt;         &lt;sp:MustSupportIssuedTokens/&gt;         &lt;sp:RequireClientEntropy/&gt;         &lt;sp:RequireServerEntropy/&gt;       &lt;/wsp:Policy&gt;     &lt;/sp:Trust10&gt;   &lt;/wsp:Policy&gt; &lt;/sp:BootstrapPolicy&gt; &lt;/wsp:Policy&gt; &lt;/sp:SecureConversationToken&gt; &lt;/wsp:Policy&gt; &lt;/sp:ProtectionToken&gt; &lt;sp:AlgorithmSuite&gt;   &lt;wsp:Policy&gt;     &lt;sp:Basic256/&gt;   &lt;/wsp:Policy&gt; &lt;/sp:AlgorithmSuite&gt; &lt;sp:Layout&gt;   &lt;wsp:Policy&gt;     &lt;sp:Strict/&gt;   &lt;/wsp:Policy&gt; &lt;/sp:Layout&gt; &lt;sp:IncludeTimestamp/&gt; &lt;sp:EncryptSignature/&gt; &lt;sp:OnlySignEntireHeadersAndBody/&gt; &lt;/wsp:Policy&gt; &lt;/sp:SymmetricBinding&gt; &lt;sp:Wss11 xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy"&gt;   &lt;wsp:Policy&gt;     &lt;sp:MustSupportRefKeyIdentifier/&gt;     &lt;sp:MustSupportRefIssuerSerial/&gt;     &lt;sp:MustSupportRefThumbprint/&gt;     &lt;sp:MustSupportRefEncryptedKey/&gt;   &lt;/wsp:Policy&gt; &lt;/sp:Wss11&gt; &lt;sp:Trust10 xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy"&gt;   &lt;wsp:Policy&gt;     &lt;sp:MustSupportIssuedTokens/&gt;     &lt;sp:RequireClientEntropy/&gt;     &lt;sp:RequireServerEntropy/&gt;   &lt;/wsp:Policy&gt; &lt;/sp:Trust10&gt; &lt;wsaw:UsingAddressing/&gt; &lt;/wsp:All&gt; &lt;/wsp:ExactlyOne&gt; &lt;/wsp:Policy&gt; </pre>
8.	<p>After the XML Policy is saved, select the MessageSecurityWithUsernameClient "Aggregate Policy" in the Policies &gt; Organizational Polices folder of the Organization Tree. The "Aggregate Policy Details" screen displays.</p>
9.	<p>In the "Aggregate Policy" portlet, click "Add." The "Add Policy Wizard" launches and the "Select Policy Creation Option" screen displays. Click the "Add Policy" radio button.</p>

### To Configure Message Security with a User Name Client

	From the "Type" drop-down list box select "WS-Security Message Policy." Click <b>Next</b> . On the "Specify Policy Details" screen enter the policy "Name" (e.g., MessageSecurityWithUsernameClient-Message) and optional "Description."
10.	To save the policy click <b>Finish</b> then <b>Close</b> . The "Policy Details" screen for the "XML Policy" displays.
11.	After these steps are complete the MessageSecurityWithUsernameClient" Aggregate Policy" will include two policies as shown below: <div style="text-align: center;">  </div> <p style="text-align: center;"><b>Figure 3-4: MessageSecurityWithUsernameClient—via Organization Tree</b></p>

The following procedure illustrates how to attach a policy to a service endpoint.

### To Attach Policy to Service Endpoint

Step	Procedure
1.	To attach a new policy to a service endpoint, in the Organization Tree of the Policy Manager "Management Console" select the service you would like to attach a policy to. The "Service Details" screen displays.
2.	Select the "Bindings" tab. Click on the binding "Qualified Name." The "Binding Details" screen displays. In the "Operational Policy Attachments" portlet click "Manage." The "Manage Operational Policy Attachments" screen displays.
3.	Navigate the tree hierarchy, and click the checkbox of one or more policies you would like to attach to the current binding. In this example, select the MessageSecurityWithUsernameClient Aggregate Policy. Click <b>Apply</b> . The selected policies are added to the "Manage Operational Policy Attachments" screen.

### Testing Service

Use the service metadata URL ("WSDL URL" link) on the "Service Overview" portlet to generate a proxy, sample messages and configurations relevant to your external tool. Confirm that the service is configured with service behavior that defines a service certificate, for example:

```
<behavior name="ServiceBehaviorWithCertificate">
  <serviceCredentials>
    <serviceCertificate findValue="00000000000000000000 0000000000000000"
      storeLocation="LocalMachine"
      storeName="My"
      x509FindType="FindByThumbprint" />
  </serviceCredentials>
</behavior>
</serviceBehaviors>
```



If the service is managed by the Agent for WCF, this behavior is automatically added to the service definition.

## TRANSPORT SECURITY WITH WINDOWS AUTHENTICATION OVER HTTPS

This scenario is using wsHttpBinding to secure client and service applications deployed in a domain with Kerberos controller over HTTPS transport.

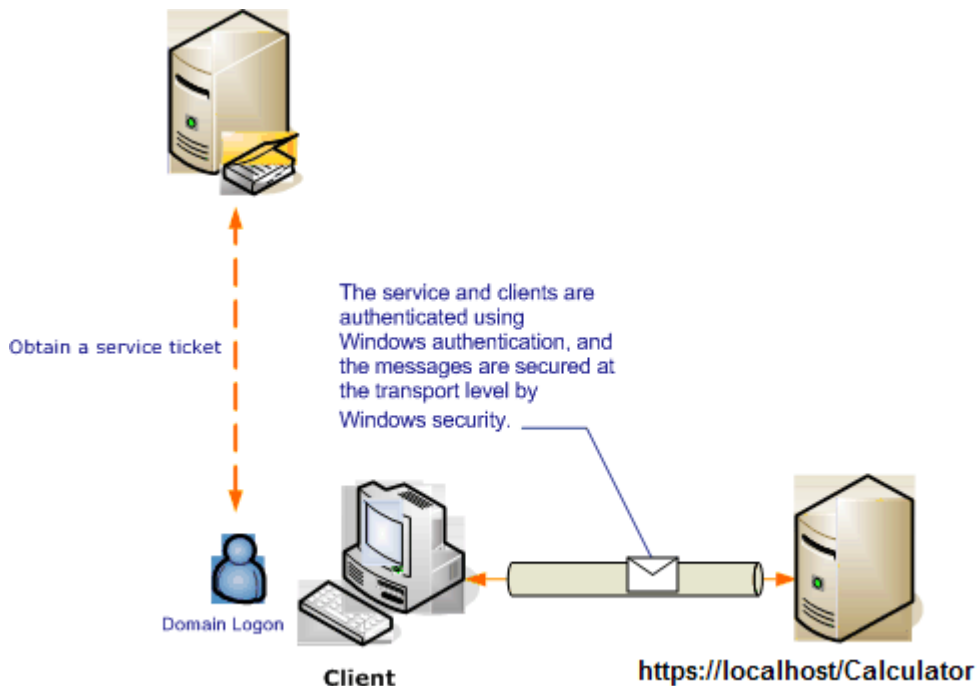


Figure 3-5: Transport Security with Windows Authentication over HTTPS

### Interoperability

Windows Communication Foundation Only

### WCF Configuration

```
<wsHttpBinding>
  <binding name="TransportSecurityWithWindowsAuthenticationOverHttps">
    <security mode="Transport">
      <transport clientCredentialType="Windows" />
    </security>
  </binding>
</wsHttpBinding>
```

## Policy Manager Configuration

The following procedure illustrates how to configure Transport Security with Windows Authentication over HTTPS.

### To Configure Transport Security with Windows Authentication over HTTPS

Step	Procedure
1.	In the Organization Tree of the Policy Manager "Management Console" select Organization > Policies Folder > Operational Tab.
2.	Click "Add Policy." The "Add Policy Wizard" launches and the "Select Policy Creation Option" screen displays.
3.	Click the "Add Policy" radio button. From the "Type" drop-down list box select "Aggregate Policy." Click <b>Next</b> . On the "Specify Policy Details" screen enter the policy "Name" (e.g., TransportSecurityWithWindowsAuthenticationOverHttps) and optional "Description."
4.	To save the policy click <b>Finish</b> then <b>Close</b> . The "Policy Details" screen for the Aggregate Policy displays.
5.	In the "Aggregate Policy" portlet, click "Add." The "Add Policy Wizard" launches and the "Select Policy Creation Option" screen displays. Click the "Add Policy" radio button. From the "Type" drop-down list box select "XML Policy." Click <b>Next</b> . On the "Specify Policy Details" screen enter the policy "Name" (e.g., Binding Policy) and optional "Description."
6.	To save the policy click <b>Finish</b> then <b>Close</b> . The "Policy Details" screen for the "XML Policy" displays.
7.	In the "XML Policy" portlet, click "Modify." The "Modify XML Policy" screen displays. In the "XML Policy Content" text box insert the following XML Policy definition and click <b>Apply</b> to save the policy definition: <pre> &lt;wsp:Policy wsu:Id="WSHttpBinding_TransportSecurityWithWindowsAuthenticationOverHttps"     xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"     xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"     xmlns:wsaw="http://www.w3.org/2006/05/addressing/wSDL"   &gt;   &lt;wsp:ExactlyOne&gt;     &lt;wsp&gt;All&gt;       &lt;http:NegotiateAuthentication xmlns:http="http://schemas.microsoft.com/ws/06/2004/policy/http"/&gt;       &lt;sp:TransportBinding xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy"&gt;         &lt;wsp:Policy&gt;           &lt;sp:TransportToken&gt;             &lt;wsp:Policy&gt;               &lt;sp:HttpsToken RequireClientCertificate="false"/&gt;             &lt;/wsp:Policy&gt;           &lt;/sp:TransportToken&gt;           &lt;sp:AlgorithmSuite&gt;             &lt;wsp:Policy&gt;               &lt;sp:Basic256/&gt;             &lt;/wsp:Policy&gt;           &lt;/sp:AlgorithmSuite&gt;         &lt;/wsp:Policy&gt;       &lt;/sp:TransportBinding&gt;     &lt;/wsp&gt;All&gt;   &lt;/wsp:ExactlyOne&gt; </pre>

### To Configure Transport Security with Windows Authentication over HTTPS

	<pre> &lt;/wsp:Policy&gt; &lt;/sp:AlgorithmSuite&gt; &lt;sp:Layout&gt;   &lt;wsp:Policy&gt;     &lt;sp:Strict/&gt;   &lt;/wsp:Policy&gt; &lt;/sp:Layout&gt; &lt;/wsp:Policy&gt; &lt;/sp:TransportBinding&gt; &lt;wsaw:UsingAddressing/&gt; &lt;/wsp:All&gt; &lt;/wsp:ExactlyOne&gt; &lt;/wsp:Policy&gt; </pre>
--	--

The following procedure illustrates how to attach a policy to a service endpoint.

### To Attach Policy to Service Endpoint

Step	Procedure
1.	To attach a new policy to a service endpoint, in the Organization Tree of the Policy Manager "Management Console" select the service you would like to attach a policy to. The "Service Details" screen displays.
2.	Select the "Bindings" tab. Click on the binding "Qualified Name." The "Binding Details" screen displays. In the "Operational Policy Attachments" portlet click "Manage." The "Manage Operational Policy Attachments" screen displays.
3.	Navigate the tree hierarchy and click the checkbox of one or more policies you would like to attach to the current binding. In this example, select the XML Policy you just created. Click <b>Apply</b> . The selected policies are added to the "Manage Operational Policy Attachments" screen.
4.	Confirm that the service Access Point address is defined with HTTPS Transport Schema. If it is not, click the "Access Points" tab of the current service. For the Access Point line item select "Modify Access Point" from the "Actions" drop-down list box. On the "Specify Access Point Details" screen click <b>Next</b> . The details page for the Access Point binding displays. Modify the endpoint address and click <b>Finish</b> .

### Testing Service

If your physical service is managed by the Agent for WCF and it is tested by the WCF client application managed by the SOA Software WCF Delegate, reconfiguring with either a service or client application is not required as they are both managed by SOA Software. Note: if a service is deployed in IIS Server, confirm that Basic Authentication is enabled for the virtual directory that hosts your service. If Basic Authentication is not enabled, the WCF service will throw an exception at startup.

If you are testing a service with an external tool that is not managed by the SOA Software Delegate, use the service metadata URL ("WSDL URL" link) on the "Service Overview" portlet to generate a proxy, sample messages, and configurations relevant to your external tool.

## MESSAGE SECURITY WITH A CERTIFICATE CLIENT

This scenario shows a Windows Communication Foundation (WCF) client and service secured using message security mode. Both the client and the service are authenticated with certificates.

### Interoperability

Windows Communication Foundation Only

### WCF Configuration

```
<wsHttpBinding>
  <binding name="MutualCertificateBinding">
    <security mode="Message">
      <message clientCredentialType="Certificate"/>
    </security>
  </binding>
</wsHttpBinding>
```

### Policy Manager Configuration

The following procedure illustrates how to configure Message Security with a Certificate Client.

#### To Configure Message Security with a Certificate Client

Step	Procedure
1.	In the Organization Tree of the Policy Manager "Management Console," select Organization > Policies Folder > Operational Tab.
2.	Click "Add Policy." The "Add Policy Wizard" launches and the "Select Policy Creation Option" screen displays.
3.	Click the "Add Policy" radio button. From the "Type" drop-down list box select "Aggregate Policy." Click <b>Next</b> . On the "Specify Policy Details" screen enter the policy "Name" (e.g., MutualSecurityWithCertificateClient) and optional "Description."
4.	To save the policy click <b>Finish</b> then <b>Close</b> . The "Policy Details" screen for the Aggregate Policy displays.
5.	In the "Aggregate Policy" portlet, click "Add." The "Add Policy Wizard" launches and the "Select Policy Creation Option" screen displays. Click the "Add Policy" radio button. From the "Type" drop-down list box select "XML Policy." Click <b>Next</b> . On the "Specify Policy Details" screen enter the policy "Name" (e.g., MutualSecurityWithCertificateClient-Binding) and optional "Description."
6.	To save the policy click <b>Finish</b> then <b>Close</b> . The "Policy Details" screen for the "XML Policy" displays.
7.	In the "XML Policy" portlet, click "Modify." The "Modify XML Policy" screen displays. In the "XML Policy Content" text box insert the following XML Policy definition and click <b>Apply</b> to save the policy definition: <pre>&lt;wsp:Policy wsu:Id="WSHttpBinding_MutualCertificateBinding"   xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-</pre>

## To Configure Message Security with a Certificate Client

```

200401-wss-wssecurity-utility-1.0.xsd"
    xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
  >
    <wsp:ExactlyOne>
      <wsp>All>
        <sp:SymmetricBinding
xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
          <wsp:Policy>
            <sp:ProtectionToken>
              <wsp:Policy>
                <sp:SecureConversationToken
sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/Inc
ludeToken/AlwaysToRecipient">
                  <wsp:Policy>
                    <sp:RequireDerivedKeys/>
                    <sp:BootstrapPolicy>
                      <wsp:Policy>
                        <sp:SignedParts>
                          <sp:Body/>
                          <sp:Header Name="To"
Namespace="http://www.w3.org/2005/08/addressing" />
                          <sp:Header Name="From"
Namespace="http://www.w3.org/2005/08/addressing" />
                          <sp:Header Name="FaultTo"
Namespace="http://www.w3.org/2005/08/addressing" />
                          <sp:Header Name="ReplyTo"
Namespace="http://www.w3.org/2005/08/addressing" />
                          <sp:Header Name="MessageID"
Namespace="http://www.w3.org/2005/08/addressing" />
                          <sp:Header Name="RelatesTo"
Namespace="http://www.w3.org/2005/08/addressing" />
                          <sp:Header Name="Action"
Namespace="http://www.w3.org/2005/08/addressing" />
                        </sp:SignedParts>
                        <sp:EncryptedParts>
                          <sp:Body/>
                        </sp:EncryptedParts>
                      <sp:SymmetricBinding>
                        <wsp:Policy>
                          <sp:ProtectionToken>
                            <wsp:Policy>
                              <mssp:SslContextToken
sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/Inc
ludeToken/AlwaysToRecipient"
xmlns:mssp="http://schemas.microsoft.com/ws/2005/07/securitypolicy">
                                <wsp:Policy>
                                  <sp:RequireDerivedKeys/>
                                  <mssp:RequireClientCertificate/>
                                </wsp:Policy>
                              </mssp:SslContextToken>
                            </wsp:Policy>
                          </sp:ProtectionToken>
                        <sp:AlgorithmSuite>
                          <wsp:Policy>
                            <sp:Basic256/>
                          </wsp:Policy>
                        </sp:AlgorithmSuite>
                      <sp:Layout>
                        <wsp:Policy>
                          <sp:Strict/>
                        </wsp:Policy>
                      </sp:SymmetricBinding>
                    </wsp:Policy>
                  </sp:SecureConversationToken>
                </wsp:Policy>
              </sp:ProtectionToken>
            </wsp:Policy>
          </sp:ProtectionToken>
        </wsp:Policy>
      </wsp>All>
    </wsp:ExactlyOne>
  </wsp:Policy>

```

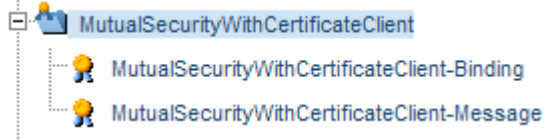
## To Configure Message Security with a Certificate Client

```

        </wsp:Policy>
    </sp:Layout>
    <sp:IncludeTimestamp/>
    <sp:EncryptSignature/>
    <sp:OnlySignEntireHeadersAndBody/>
</wsp:Policy>
</sp:SymmetricBinding>
<sp:Wss11>
    <wsp:Policy>
        <sp:MustSupportRefKeyIdentifier/>
        <sp:MustSupportRefIssuerSerial/>
        <sp:MustSupportRefThumbprint/>
        <sp:MustSupportRefEncryptedKey/>
    </wsp:Policy>
</sp:Wss11>
<sp:Trust10>
    <wsp:Policy>
        <sp:MustSupportIssuedTokens/>
        <sp:RequireClientEntropy/>
        <sp:RequireServerEntropy/>
    </wsp:Policy>
</sp:Trust10>
</wsp:Policy>
</sp:BootstrapPolicy>
</wsp:Policy>
</sp:SecureConversationToken>
</wsp:Policy>
</sp:ProtectionToken>
<sp:AlgorithmSuite>
    <wsp:Policy>
        <sp:Basic256/>
    </wsp:Policy>
</sp:AlgorithmSuite>
<sp:Layout>
    <wsp:Policy>
        <sp:Strict/>
    </wsp:Policy>
</sp:Layout>
    <sp:IncludeTimestamp/>
    <sp:EncryptSignature/>
    <sp:OnlySignEntireHeadersAndBody/>
</wsp:Policy>
</sp:SymmetricBinding>
<sp:Wss11
xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
    <wsp:Policy>
        <sp:MustSupportRefKeyIdentifier/>
        <sp:MustSupportRefIssuerSerial/>
        <sp:MustSupportRefThumbprint/>
        <sp:MustSupportRefEncryptedKey/>
    </wsp:Policy>
</sp:Wss11>
<sp:Trust10
xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
    <wsp:Policy>
        <sp:MustSupportIssuedTokens/>
        <sp:RequireClientEntropy/>
        <sp:RequireServerEntropy/>
    </wsp:Policy>
</sp:Trust10>
<wsaw:UsingAddressing/>

```

### To Configure Message Security with a Certificate Client

	<pre>&lt;/wsp:All&gt; &lt;/wsp:ExactlyOne&gt; &lt;/wsp:Policy&gt;</pre>
8.	After the XML Policy is saved, select the MutualSecurityWithCertificateClient "Aggregate Policy" in the Policies > Organizational Polices folder of the Organization Tree. The "Aggregate Policy Details" screen displays.
9.	In the "Aggregate Policy" portlet, click "Add." The "Add Policy Wizard" launches and the "Select Policy Creation Option" screen displays. Click the "Add Policy" radio button. From the "Type" drop-down list box select "WS-Security Message Policy." Click <b>Next</b> . On the "Specify Policy Details" screen enter the policy "Name" (e.g., MutualSecurityWithCertificateClient Policy) and optional "Description."
10.	To save the policy click <b>Finish</b> then <b>Close</b> . The "Policy Details" screen for the "WS-Security Message Policy" displays.
11.	After these steps are complete the MutualSecurityWithCertificateClient" Aggregate Policy" will include two policies as shown below: <div style="text-align: center;">  </div> <p style="text-align: center;"><b>Figure 3-6: MutualSecurityWithCertificateClient—via Organization Tree</b></p>

The following procedure illustrates how to attach a policy to a service endpoint.

### To Attach Policy to Service Endpoint

Step	Procedure
1.	To attach a new policy to a service endpoint, in the Organization Tree of the Policy Manager "Management Console" select the service you would like to attach a policy to. The "Service Details" screen displays.
2.	Select the "Bindings" tab. Click on the binding "Qualified Name." The "Binding Details" screen displays. In the "Operational Policy Attachments" portlet click "Manage." The "Manage Operational Policy Attachments" screen displays.
3.	Navigate the tree hierarchy and click the checkbox of one or more policies you would like to attach to the current binding. In this example, select the XML Policy you just created. Click <b>Apply</b> . The selected policies are added to the "Manage Operational Policy Attachments" screen.

## Testing Service

Use the service metadata URL ("WSDL URL" link) from the "Service Overview" portlet to generate a proxy, sample messages, and configurations relevant to your external tool. Confirm that the service is configured with service behavior that defines a service certificate, for example:

```
<behavior name="ServiceBehaviorWithCertificate">
  <serviceCredentials>
    <serviceCertificate findValue="00000000000000000000 000000000000000000"
      storeLocation="LocalMachine"
      storeName="My"
      x509FindType="FindByThumbprint" />
    </serviceCredentials>
  </behavior>
</serviceBehaviors>
```

If the service is managed by the Agent for WCF this behavior is automatically added to service definition.

## CONFIGURING OTHER WCF SCENARIOS

Policy Manager can be pre-configured to support other WCF use case scenarios. To create a base policy to support other WCF binding configurations, review the WSDL documents produced by a service deployed with the desired WCF binding configuration. A WCF service WSDL can include a maximum of two types of `wsp:Policy` root elements. One of these root elements is applied at the binding level and other(s) at the binding message level.

- If the WCF service includes both policy element types, create an Aggregate Policy in the Policy Manager "Management Console" and add an XML policy and WS-Security Message Policy.
- If the WCF service includes one `wsp:Policy` root element, create a single XML Policy.

Configure the XML Policy with the content of the root `wsp:Policy` element found in the service WSDL that is applied at the binding level. Then apply the new Aggregate Policy to the service binding in the Policy Manager "Management Console."



# Chapter 4: Configuring WCF Policies with Network Director

## OVERVIEW

Microsoft WCF services can be accessed by WCF client applications through the SOA Software Network Director. Services exposed to WCF client applications through Network Director are called virtual services. This section describes typical policy configurations supported by Policy Manager when WCF services are virtualized through Network Director. When a virtual service is created in the Policy Manager "Management Console" and is hosted in one of the registered Network Director containers, two distinct policies will be associated with the virtual service:

- Virtual Service Outbound Policy—Policy of the physical service endpoint that virtual service is calling (virtualizing).
- Virtual Service Inbound Policy—Policy of the Network Director virtual service endpoint. This is the endpoint client applications are using to communicate with the virtual service.

Policy Manager is shipped with a series of pre-configured sample policies that can be used with WCF services (Refer to the "Managing WCF Services with Policy Manager Security Policies" chapter). When these policies are considered in the context of Network Director and virtual services, they will be applied as both virtual service Outbound and Inbound Policies. This allows the policy of the virtual service endpoint to be different from the policy of the physical service that is virtualized.

Note: Each procedure assumes you are logged into the Policy Manager "Management Console" and have the Policy Manager "Workbench" tab selected.

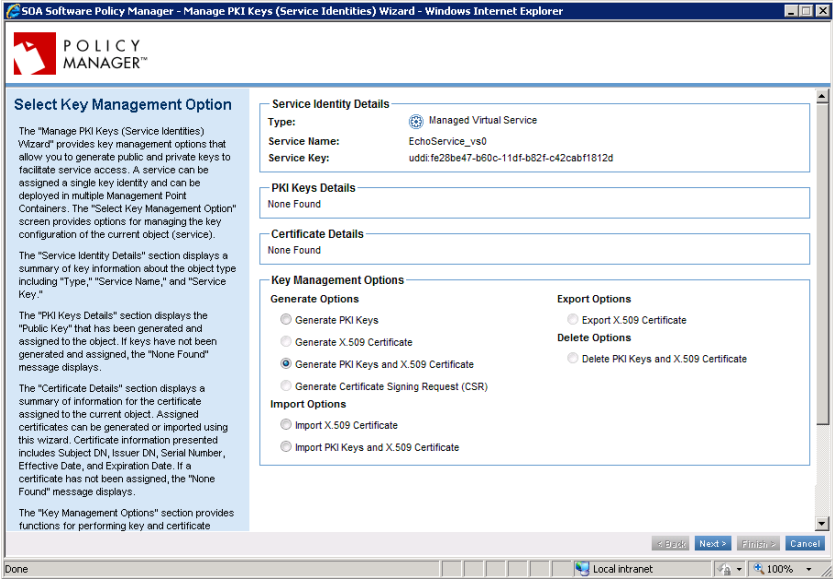
## CONFIGURING VIRTUAL SERVICE INBOUND POLICIES

### **Policies: AnonymousForCertificate, MutualCertificateSignEncrypt, MutualCertificateSignOnly, MutualCertificateSymmetricBinding, and UsernameForCertificate**

The AnonymousForCertificate, MutualCertificateSignEncrypt, MutualCertificateSignOnly, MutualCertificateSymmetricBinding, and UsernameForCertificate policies require that a virtual service be configured with an X.509 certificate.

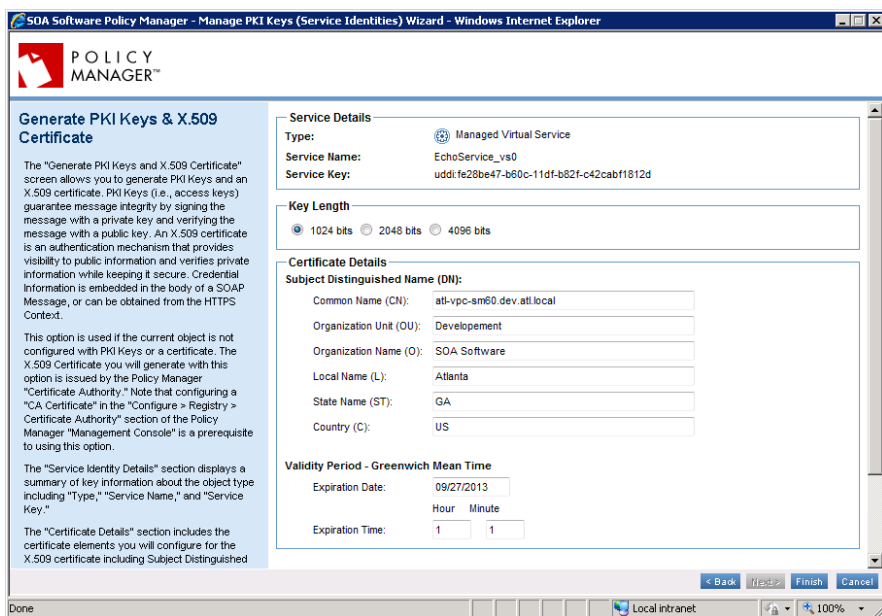
To configure a virtual service with an X.509 certificate perform the following steps:

### To Configure Virtual Service with X.509 Certificate

Step	Procedure
1.	In the Organization Tree, click the "Services" folder and select a virtual service. The "Service Details" screen displays.
2.	In the "Actions" portlet, click "Manage PKI Keys." The "Select Key Management Option" screens displays.
3.	<p>To configure a virtual service with an X.509 certificate the following options can be used:</p> <ul style="list-style-type: none"> <li>• Generate PKI Keys with X.509 Certificate—Used to generate a new certificate.</li> <li>• Import X.509 Certificate—Used to import an existing certificate.</li> </ul>
4.	<p><b>To generate new certificate:</b></p> <p>Select the "Generate PKI Keys and X.509 Certificate" radio button and click <b>Next</b> to continue.</p>  <p>The screenshot shows a web browser window titled "SOA Software Policy Manager - Manage PKI Keys (Service Identities) Wizard - Windows Internet Explorer". The page has a "POLICY MANAGER" logo and a "Select Key Management Option" heading. It contains several sections: "Service Identity Details" (Type: Managed Virtual Service, Service Name: EchoService_vs0, Service Key: uddl:fe28be47-b60c-11df-b82f-c42cabf1812d), "PKI Keys Details" (None Found), "Certificate Details" (None Found), "Key Management Options" (with sub-sections for Generate, Import, Export, and Delete), and "Import Options". Under "Generate Options", the radio button for "Generate PKI Keys and X.509 Certificate" is selected. At the bottom, there are "Back", "Next", "Finish", and "Cancel" buttons.</p>
5.	The "Generate PKI Keys & X.509 Certificate" screen displays.

**Figure 4-1: Manage PKI Keys (Service Identities)—Select Key Management Option (Generate PKI Keys and X.509 Certificate)**

## To Configure Virtual Service with X.509 Certificate

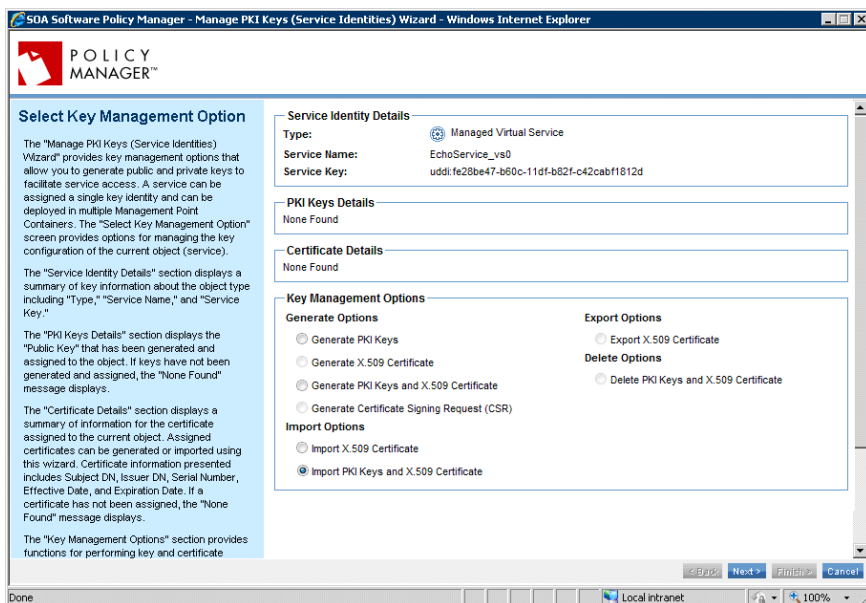


**Figure 4-2: Manage PKI Keys (Service Identities)—Generate PKI Keys & X.509 Certificate**

Select "Key Length," and specify "Certificate Details" and "Validity Period." To generate the certificate, click **Finish**.

6. **To import existing certificate:**

Select the "Import PKI Keys and X.509 Certificate" radio button and click **Next** to continue.



**Figure 4-3: Manage PKI Keys (Service Identities)—Select Key Management Option (Import PKI Keys and X.509 Certificate)**

## To Configure Virtual Service with X.509 Certificate

7. The "Import Private Key & X.509 Certificate from Keystore" screen displays.

**Figure 4-4: Manage PKI Keys (Service Identities)—Import Private Key & X.509 Certificate from Keystore**

Configure the "Keystore Details" as follows:

- Keystore Type—Click the "PKCS12" radio button.
- Keystore Path—To specify the file location of the X.509 Certificate, click "Browse." The "Choose File" dialog displays. Navigate to the directory location where the certificate is stored, select the certificate file and click Open. The "Keystore Path" is populated with the directory location of the X.509 Certificate.
- Keystore Password—Enter the "Keystore Password" for the certificate and confirm it.
- Key Alias— Click "Load Aliases." The "Key Alias" drop-down list box is populated with a list of Key Alias entries that are defined in the specified keystore file. Select the Key Alias you would like to import from the drop-down list box. Enter the Key Password and confirm it.

To import the Private Key and X.509 Certificate, click **Finish**. The certificate is imported into the Policy Manager data repository.

## **Policies: CertificateOverTransport, KerberosOverTransport and UsernameOverTransport**

The CertificateOverTransport, KerberosOverTransport and UsernameOverTransport policies require HTTPS transport (and endpoints) to be configured with the virtual service.

## **CONFIGURING VIRTUAL SERVICE OUTBOUND POLICIES**

### **Policies: AnonymousForCertificate, MutualCertificateSignEncrypt, MutualCertificateSignOnly, MutualCertificateSymmetricBinding, and UsernameForCertificate**

The AnonymousForCertificate, MutualCertificateSignEncrypt, MutualCertificateSignOnly, MutualCertificateSymmetricBinding, and UsernameForCertificate policies require physical service to be configured with X.509 certificate. Use the same procedures described above for virtual services to assign X.509 certificate to a physical service. You cannot use "Generate PKI Keys and X.509 Certificate" because an option for exporting a generated certificate along with its private key is not available.

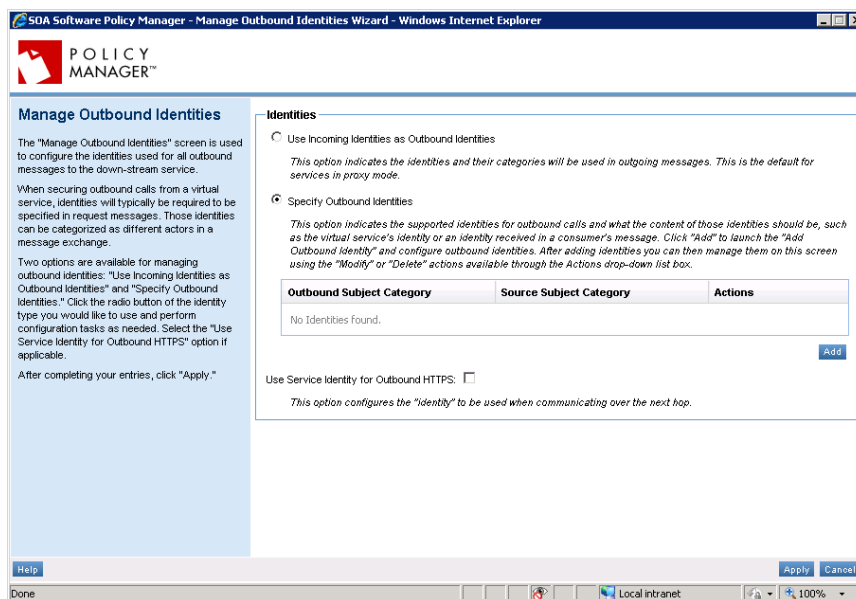
### **Policies: CertificateOverTransport, MutualCertificateSignEncrypt, MutualCertificateSignOnly and MutualCertificateSymmetricBinding**

The CertificateOverTransport, MutualCertificateSignEncrypt, MutualCertificateSignOnly and MutualCertificateSymmetricBinding policies require that a virtual service to be configured with an X.509 outbound identity.

#### **To Configure Virtual Service Outbound Policies (Specify Outbound Identities)**

<b>Step</b>	<b>Procedure</b>
1.	In the Organization Tree, click the "Services" folder and select a virtual service. The "Service Details" screen displays.
2.	In the "Actions" portlet, click "Manage Outbound Identities." The "Manage Outbound Identities Wizard" launches and the "Manage Outbound Identities" screen displays.

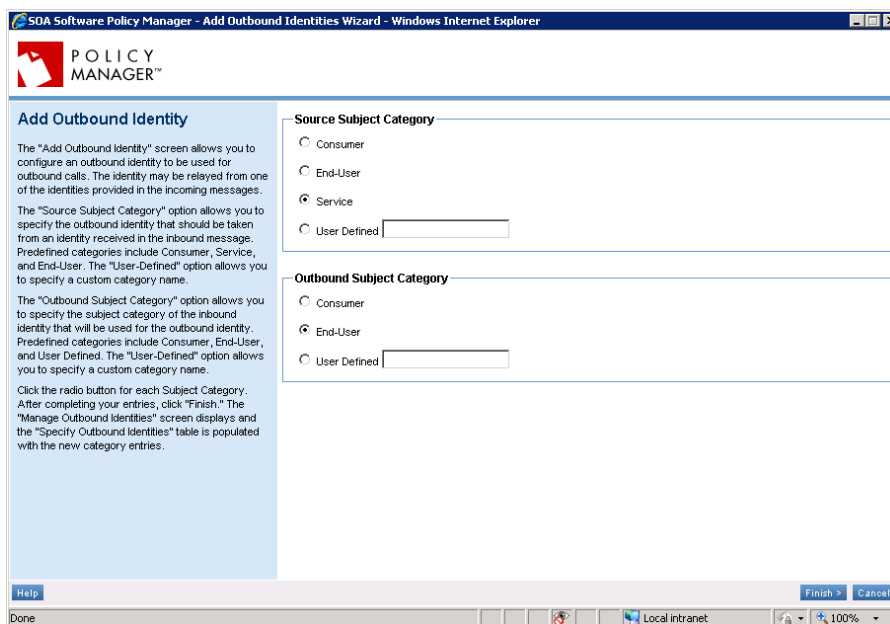
## To Configure Virtual Service Outbound Policies (Specify Outbound Identities)



**Figure 4-5: Manage Outbound Identities Wizard—Manage Outbound Identities (Specify Outbound Identities selected)**

Perform the following steps:

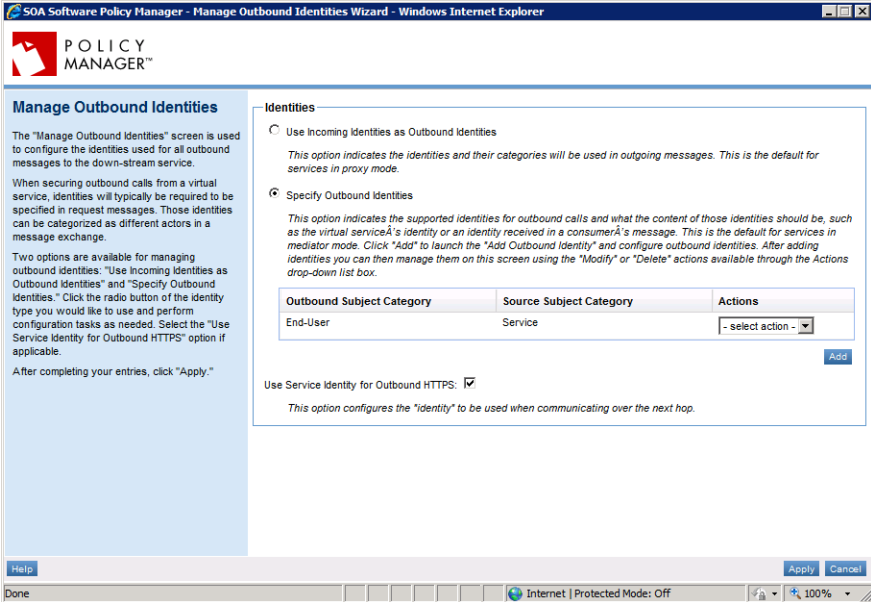
- Click the "Specify Outbound Identities" radio button.
- In the "Specify Outbound Identities" section, click "Add." The "Add Outbound Identity" screen displays.



**Figure 4-6: Manage Outbound Identities Wizard—Add Outbound Identity**

- In the "Source Subject Category" section click the "Service" radio button.

### To Configure Virtual Service Outbound Policies (Specify Outbound Identities)

	<ul style="list-style-type: none"> <li>• In the "Outbound Subject Category" section click the "End-User" radio button.</li> </ul>
<p>3.</p>	<p>Click <b>Finish</b> to continue. The "Manage Outbound Identities" screen displays. The "Specify Outbound Identities" radio button is selected. Click the "Use Service Identity for Outbound HTTPS" checkbox. Click <b>Apply</b>. The "Manage Outbound Identities Wizard" closes and saves the identity configuration.</p>  <p><b>Figure 4-7: Manage Outbound Identities Wizard—Manage Outbound Identities (Specify Outbound Identities configured)</b></p>

### Policies: UsernameForCertificate and UsernameOverTransport

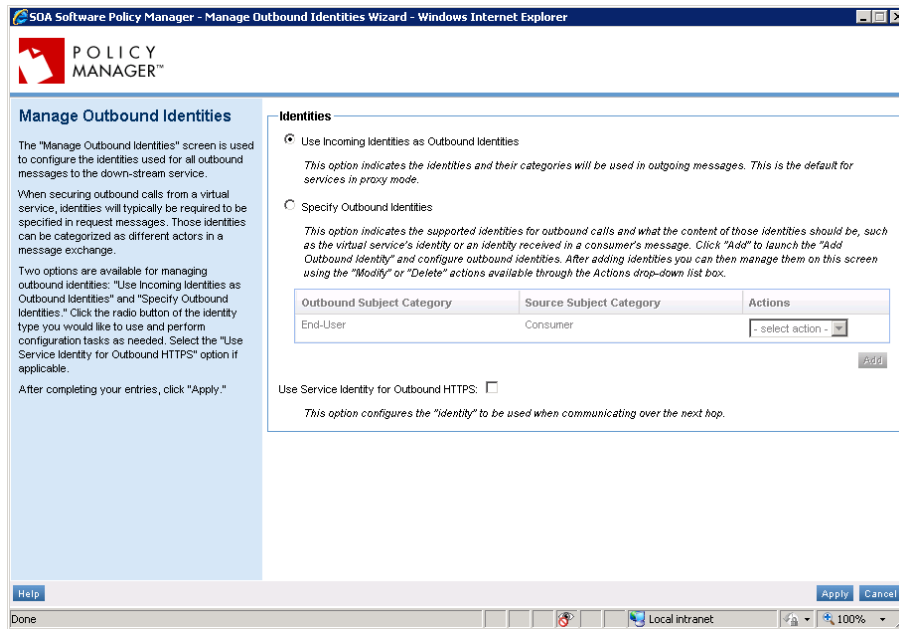
The UsernameForCertificate and UsernameOverTransport policies require that a username token be provided by the virtual service when it calls a physical service. Network Director can use a username/password it receives on its inbound side as its impersonated identity that is used on its outbound side. Impersonation is enabled by selecting the "Use Incoming Identities as Outbound Identities" option in the Manage Outbound Identities screen.

### To Configure Virtual Service Outbound Policies (Use Incoming Identities as Outbound Identities)

Step	Procedure
1.	In the Organization Tree, click the "Services" folder and select a virtual service. The "Service Details" screen displays.
2.	In the "Actions" portlet, click "Manage Outbound Identities." The "Manage Outbound Identities Wizard" launches and the "Manage Outbound Identities" screen displays.
3.	Click the "Use Incoming Identities as Outbound Identities" radio button. Click <b>Apply</b> .

## To Configure Virtual Service Outbound Policies (Use Incoming Identities as Outbound Identities)

The "Manage Outbound Identities Wizard" closes and saves the identity configuration.



**Figure 4-8: Manage Outbound Identities Wizard—Manage Outbound Identities (Use Incoming Identities as Outbound Identities selected)**