



# **FORUM SENTRY™ VERSION 9**

## **WS-TRUST CLIENT CONFIGURATION GUIDE**

**Legal Marks**

No portion of this document may be reproduced or copied in any form, or by any means – graphic, electronic, or mechanical, including photocopying, taping, recording, or information retrieval system – without expressed permission from Forum Systems, Inc.

FORUMOS™ Firmware, Forum Systems XMLSec™ WebAdmin, Forum Systems XML Security Appliance™, Forum Sentry™, Forum Presidio™, Forum XWall™, Forum Sentry™ Web Services Gateway, Forum Presidio™ OpenPGP Gateway, Forum FIA Gateway™, Forum XWall Type-PCI™, Forum XWall® Web Services Firewall and Forum XRay™ are trademarks and registered trademarks of Forum Systems, Inc.

All other products are trademarks or registered trademarks of their respective companies.

Copyright © 2002-2024 Forum Systems, Inc. – All Rights Reserved.

Forum Systems Sentry™ Version 9 WS-Trust Integration Guide, published May 2024.

D-ASF-SE-828506

## Contents

Contents .....	3
INTRODUCTION TO THE WS-TRUST INTEGRATION GUIDE .....	3
<i>Audience for the WS-Trust Integration Guide</i> .....	3
<i>Conventions Used</i> .....	3
<i>Assumptions</i> .....	4
WS-TRUST USER POLICIES .....	5
<i>Authentication and Authorization using WS-Trust Policies</i> .....	5
<i>Persistent Sessions and WS-Trust</i> .....	5
<i>Creating WS-Trust Policies</i> .....	6
<i>WS-Trust Screen Terms</i> .....	6
<i>WS-Trust Policy Terms</i> .....	6
<i>Add a Run-time WS-Trust Policy</i> .....	8
Run-time Access Control and WS-Trust Group Privileges .....	9
APPENDIX .....	11
<i>Appendix A - Constraints in WS-Trust Policies</i> .....	11
<i>Appendix B - Specifications in WS-Trust Policies</i> .....	11
<i>Appendix C - Request to WS-Trust Security Token Service</i> .....	11
<i>Appendix D - Response from WS-Trust Security Token Service for Valid Credentials</i> .....	12
<i>Appendix E - Response from WS-Trust Security Token Service for Invalid Credentials</i> .....	12
INDEX .....	13

## INTRODUCTION TO THE WS-TRUST INTEGRATION GUIDE

### Audience for the WS-Trust Integration Guide

The *Forum Systems Sentry™ Version 9 WS-Trust Integration Guide* is for System Administrators who will manage access control with WS-Trust users, groups and policies.

### Conventions Used

A red asterisk ( \* ) aligned with a field term means that this field is required. In this and other documentation, the Web Administration Interface is referred to as the WebAdmin and the Forum XML Security Appliance™ is referred to as the 'device', 'product' or 'system'.

In this document, all data or commands that must be entered or selected are displayed in boldface. Example:

User name:     **johnsmith**  
Password:     \*\*\*\*\*

UI screens which display a STATUS column represent the following states:

- Green status light = enabled policy.
- Yellow status light = a required functional element of this policy is disabled.
- Red status light = disabled policy.

Throughout this and other documents in the Documentation Set, repetitive actions such as:

- View / edit a policy.
- Enable / disable a policy.
- Delete a policy.
- Rename a policy.

- Limit display of policies with Search or Max Results fields.

are not shown. For more information, refer to the Common Operations section of the *Forum Systems Sentry™ Version 9 Web-based Administration Guide*.

Customers with plug-in licenses on the system, confirm that your WS-Trust Authentication feature is visible on the General Info screen under the SUPPORTED FEATURES section.

## **Assumptions**

This document assumes that the reader will review the appropriate chapter before performing the operations listed in this document. This document also assumes that the reader is familiar with WS-Trust.

WS-Trust components may be integrated with either the Forum Sentry hardware appliance, virtual appliance, or the software edition.

## WS-TRUST USER POLICIES

Forum Sentry provides integration with the OASIS WS-Trust 1.4, and earlier specifications for authentication credentials for runtime transactions as well as design-time WebAdmin administrators. When WS-Trust integration is configured, the system acts as a WS-Trust requestor to an external WS-Trust security token service using the direct brokered trust model: The Sentry trusts the WS-Trust security token service for authentication services, and the WS-Trust security token service vouches for the identity of the user.

To set up WS-Trust integration, first a WS-Trust security token service must be set up external to the product to provide authentication services. This can be deployed using Forum Sentry as an STS Identity Broker which provides standards-based WS-Trust API for centralized identity and session management, or you can choose another STS product. Once the WS-Trust security token service is configured, the system must be configured to communicate with the WS-Trust security token service by creating a WS-Trust policy. This WS-Trust policy will have an associated group which can then be used like any other group in the system.

Standards Supported:

- WS-Trust 1.3 (<http://docs.oasis-open.org/ws-sx/ws-trust/200512>)
- WS-Trust 1.4 (<http://docs.oasis-open.org/ws-sx/ws-trust/200802>)

## Authentication and Authorization using WS-Trust Policies

The WS-Trust policies can be used to perform authentication and authorization based on association of the policy with an Access Control List. The affiliation of a WS-Trust policy with an Access Control List will result in the credentials being passed to the WS-Trust policy whenever the ACL(s) the policy is associated with are triggered during runtime transactions. When this occurs, the credentials are passed to the target STS service in the following formats based on the inbound credentials:

- 1) If the credential includes a password, the WS-Trust call will be generated using a WS-Security username header to transmit the username and password to the target STS service
- 2) If the credential does not include a password, then the WS-Trust call will be generated using a WS-Security SAML assertion to transmit the credential information within the SAML attributes.

The response from the WS-Trust call will determine success or failure of considering the credential as valid. In the cases where the credential is valid, the response information that comes back from the WS-Trust call will contain a SAML assertion and may also optionally include HTTP headers. Any attributes found within the SAML assertion or within the HTTP header will be automatically converted into User Attributes in Sentry which can be used in mapping tasks to map any of this information to other locations of the document or header of the transaction.

## Persistent Sessions and WS-Trust

Session cookies can be configured to be used based on session management at the STS server that responds to the credentials with Session Cookie artifacts. Forum STS is among the types of STS servers that can be used to Federate identities for Single-Sign-on and persistent session token management. For cases where the WS-Trust call is used to federate identities, the **Cookie Name** field is used to extract the appropriate cookie from the inbound request to pass onto the STS server to authorize the cookie and determine whether it is still a valid authentication credential.

## Creating WS-Trust Policies

While logged in as an administrator, navigate to the WS-Trust screen in the WebAdmin, available under the Access menu category in the left-hand menu. If this option does not appear, you will need to request a new license from Forum Systems which includes the WS-Trust integration feature.

The first screen offers a list of WS-Trust policies, and the ability to add/remove/modify policies. Policies can be enabled and disabled as indicated under the status column.

**WS - TRUST**

Policy settings saved

<input type="checkbox"/>	POLICY NAME	STATUS	REMOTE POLICY NAME	REMOTE PATH
<input type="checkbox"/>	<a href="#">WSTrust_A</a>	●	BostonEast-Remote	/forumadmin
<input type="checkbox"/>	<a href="#">WSTrust_B</a>	●	Cust_FS_WSDL-Remote	/forumadmin

[Delete](#) [Enable](#) [Disable](#) [New](#)

## WS-Trust Screen Terms

The WS-Trust screen includes the following terms and definitions:

TERM	DEFINITION
Policy Name	The name of the remote policy used to connect to the WS-Trust security token service.
Status	The Status column represent the following states: <ul style="list-style-type: none"><li>Green status light = enabled policy.</li><li>Yellow status light = a required functional element of this policy is disabled.</li><li>Red status light = disabled policy.</li></ul>
Remote Policy Name	The name of the remote policy used to connect to the WS-Trust security token service
Remote Path	The path used with the remote policy to connect to the WS-Trust security token service.

## WS-Trust Policy Terms

When configuring your WS-Trust Policy Server from the WS-Trust screen, consider the following:

TERM	DEFINITION
Policy Name	The name of the WS-Trust policy

Enable privileged access	<p>Both options refer to design-time admin privileges:</p> <ul style="list-style-type: none"> <li>• With Yes selected, the user has access to the WebAdmin as a super user</li> <li>• With No selected, the administrator will have access to the WebAdmin with all Domain privileges set for the Group</li> </ul>
Restrict Menus	This option is used for design-time credential validation for WebAdmin users and enables Role restrictions.
Role Policy	When restrict menus is checked, this option sets the menu items that are displayed on login to the Web Admin interface based on the role policy.
Remote Policy	The name of the remote policy used to connect to the WS-Trust security token service
Remote Path	The path used with the remote policy to connect to the WS-Trust security token service.
Applies To	Sets the AppliesTo attribute for the SAML assertion that will get generated within the WS-Trust wrapper. This value should match what the target STS is generating for SAML assertions.
Requested Token	Format of the token. This setting should match what the target STS server is able to consume and generate. Available options are SAML 1.1 and SAML 2.0
Sign Request	Enables applying a digital signature to the request using the Signature Policy selected in the drop down menu.
Validate Issuer	Enables validation of the Issuer attribute in the returned SAML.
Validate Audience	Enables validation of the Audience attribute in the returned SAML.
Require Signature	Requires the response SAML be signed and verified against the Verification Policy selected in the drop down menu.
Require Encryption	Requires the response SAML be encrypted and decrypted using the Decryption Policy selected in the drop down menu.
Request Task List Group	Selecting a Task List Group enables task list processing of the WS-Trust request generated by Sentry.
Response Task List Group	Selecting a Task List Group enables task list processing of the WS-Trust response from the STS.
Cache Timeout	<p>Set this value to the window of time that you want to consider the previous credential result to still be valid. 30 seconds is the default value.</p> <p>Significant performance gains can be realized by setting the cache timeout value to some number of seconds where the previous credential result will be cached. This reduces the burden of having to go to the STS server for each credential validation and greatly reduces network I/O latency as a result.</p>
Cookie Name	For cases where the WS-Trust call is used to federate identities, this field is used to extract the appropriate cookie from the inbound request to pass onto the STS server to authorize the cookie and determine whether it is still a valid authentication credential.
Compatibility	This sets the OASIS WS-Trust specification mode to communicate. The recommended setting is WS-Trust.

Propagate client Host Header	When enabled propagates the HTTP Host header set by the client accessing Sentry to the STS server rather than using the Sentry set Host header.
------------------------------	---

WS-TRUST > WS-TRUST POLICY CONFIGURATION

---

WS-TRUST POLICY

---

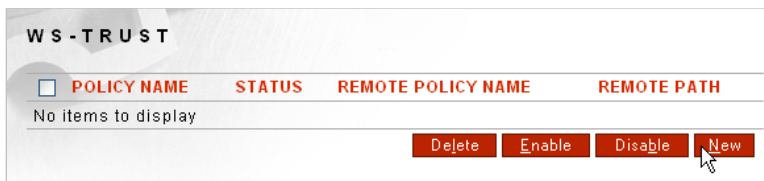
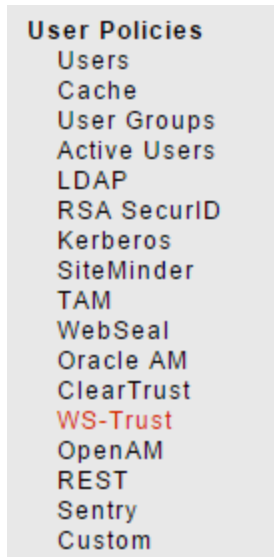
Policy Name*:	<input type="text"/>
Enable privileged access:	<input type="radio"/> Yes <input checked="" type="radio"/> No
Restrict Menus:	<input type="checkbox"/>
Role policy:	<input type="text"/>
Remote Policy:	CP_Remote_8889 <a href="#">Edit</a>
Remote Path*:	<input type="text"/>
Applies To*:	<input type="text"/>
Requested Token:	http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV1.1
Sign request:	<input checked="" type="checkbox"/>
Signature Policy:	Signature_Policy_US_DoD_test4 (RSA) <a href="#">Edit</a>
Validate issuer:	<input checked="" type="checkbox"/>
Issuer(s):	http://www.forumsys.com/sentry
Validate audience:	<input checked="" type="checkbox"/>
Audience:	http://www.forumsys.com/sentry
Require signature:	<input type="checkbox"/>
Verification Policy:	<input type="text"/>
Require encryption:	<input type="checkbox"/>
Decryption Policy:	Decryption_Policy (AES-256, 3DES) <a href="#">Edit</a>
Request Task List Group:	Task List Groups Type or select label --NONE--
Response Task List Group:	Task List Groups Type or select label --NONE--
Cache timeout (in minutes):	30
Cookie Name*:	FSSESSION
Compatibility:	<input checked="" type="radio"/> WS-Trust <input type="radio"/> Forum STS 6.x
Propagate client Host header:	<input type="checkbox"/>

[Apply](#)
[Save](#)

## Add a Run-time WS-Trust Policy

Users may create a WS-Trust policy with WS-Trust server configuration settings. This action only saves the values entered on this screen. WS-Trust groups and WS-Trust users' data are retrieved during execution time.





- From the Navigator, select the **WS-Trust** screen and select **New**.
- On the WS-TRUST POLICY CONFIGURATION screen, in the Policy Name field, enter a **name** for this WS-Trust policy.

**Note:** WS-Trust policy names must be unique and may be from 1 to 32 alphanumeric characters, may include underscores, dashes and periods. However, no trailing or leading periods are allowed.

- Select **No** for the Enable privileged access setting. This setting applies to design-time use only
- In the Remote Policy drop down list, select a **remote policy** to use to connect to the WS-Trust security token service for authentication.
- In the Remote Path field, enter a **path** to use along with the remote policy to connect to the WS-Trust security token service for authentication.
- Specify the Applies To attribute.
- Leave the Cookie Name as default.
- All other settings are optional.
- Select **Apply** or **Save**. The screen refreshes with message “Policy setting saved” visible on the screen.

**Note:** The **Apply** option retains the user on the WS-TRUST POLICY CONFIGURATION screen. The **Save** option returns the user to the WS-TRUST screen.

## Run-time Access Control and WS-Trust Group Privileges

Run-time access privilege may be set for WS-Trust policy from the ACLs screen, which is used to grant the Execute privilege. The ACL DETAILS screen displays the privileges enabled for WS-Trust policies.

The WS-Trust policy itself represents a population of users. The WS-Trust policy will appear in the ACLs screen just as a standard group will appear, and Administrators can set Execute (for run-time processing) privileges to WS-Trust policies. The following graphic displays the relationship between WS-Trust policies and run-time access control:

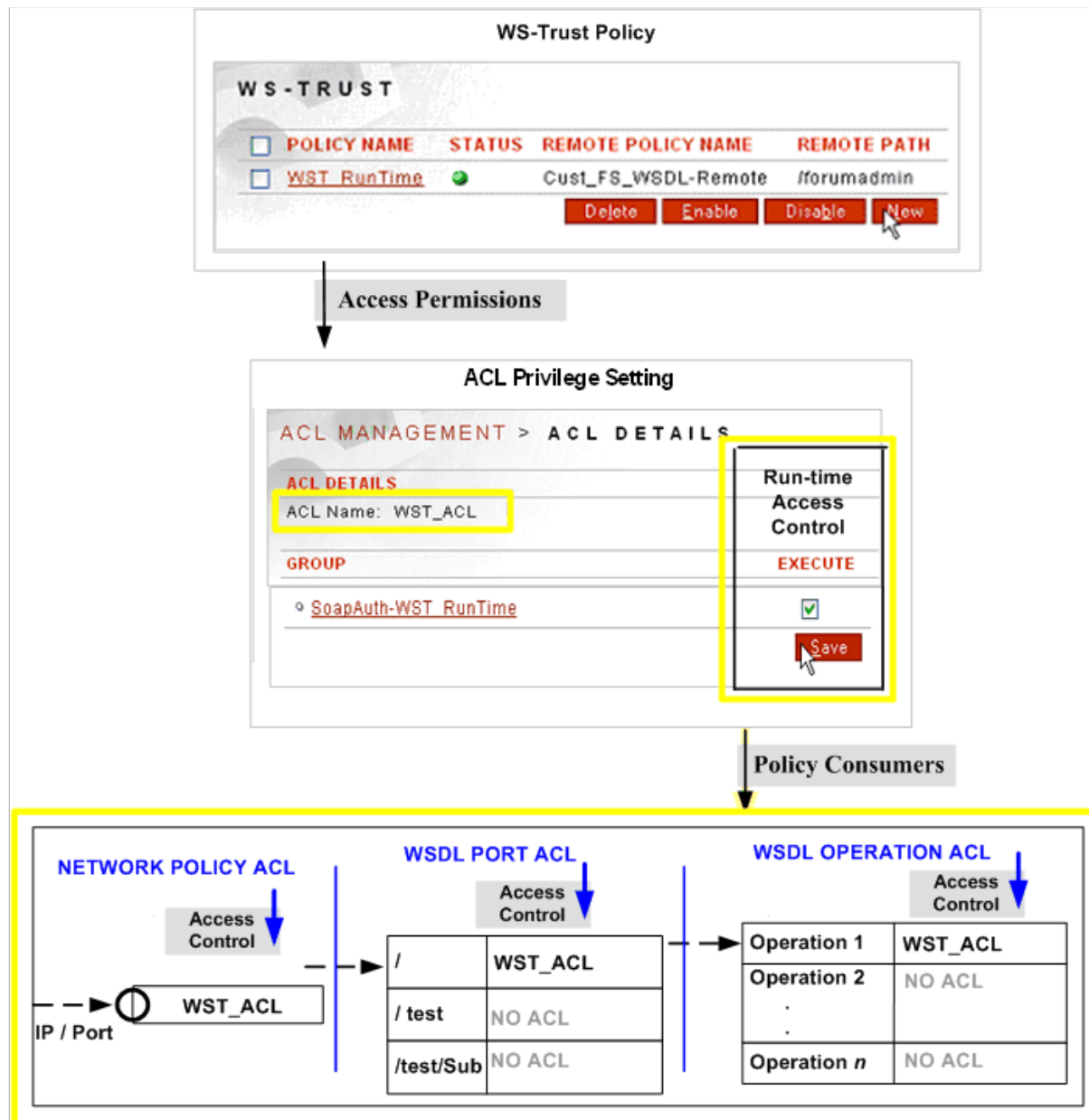


Figure 1: Run-time Access Control Defined in WS-Trust Policies by the Privileges Set in ACLs.

## APPENDIX

### Appendix A - Constraints in WS-Trust Policies

ELEMENT	CONSTRAINT	CHAR COUNT
WS-Trust Policy Name	Unique and case sensitive, may be from 1 to 32 alphanumeric characters, may include underscores, dashes and periods. However, no trailing or leading periods are allowed.	1-32

### Appendix B - Specifications in WS-Trust Policies

ELEMENT SUPPORTED	CONSTRAINT
WS-Trust Policies	Unlimited*

### Appendix C - Request to WS-Trust Security Token Service

This sample SOAP request shows what the system sends to the WS-Trust security token service to check the username user01, password pass01:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <wsse:Security
      soapenv:mustUnderstand="1"
      xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd">
      <wsse:UsernameToken
        wsu:Id="iPh6v9LBoaIGeZ9RBoR2FYZeYRvc"
        xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd">
        <wsse:Username>user01</wsse:Username>
        <wsse:Password>pass01</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <wst:RequestSecurityToken
      xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust">
      <wst:RequestType>
        http://schemas.xmlsoap.org/ws/2005/02/trust/Validate
      </wst:RequestType>
      <wst:TokenType>
        http://schemas.xmlsoap.org/ws/2005/02/trust/RSTR/Status
      </wst:TokenType>
    </wst:RequestSecurityToken>
  </soapenv:Body>
</soapenv:Envelope>
```

## Appendix D - Response from WS-Trust Security Token Service for Valid Credentials

This sample SOAP response shows what the system expects to receive from the WS-Trust security token service if the provided credentials were valid:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <wst:RequestSecurityTokenResponse
      xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust">
      <wst:TokenType>
        http://schemas.xmlsoap.org/ws/2005/02/trust/RSTR/Status
      </wst:TokenType>
      <wst:Status>
        <wst:Code>
          http://schemas.xmlsoap.org/ws/2005/02/trust/status/valid
        </wst:Code>
      </wst:Status>
    </wst:RequestSecurityTokenResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

## Appendix E - Response from WS-Trust Security Token Service for Invalid Credentials

This sample SOAP response shows what the system expects to receive from the WS-Trust security token service if the provided credentials were not valid:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <wst:RequestSecurityTokenResponse
      xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust">
      <wst:TokenType>
        http://schemas.xmlsoap.org/ws/2005/02/trust/RSTR/Status
      </wst:TokenType>
      <wst:Status>
        <wst:Code>
          http://schemas.xmlsoap.org/ws/2005/02/trust/status/invalid
        </wst:Code>
      </wst:Status>
    </wst:RequestSecurityTokenResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

## INDEX

- add run-time WS-Trust policy, 8
- conventions used, 4
- save WS-Trust server configuration settings, 8
- terms
  - in WS-Trust screen, 6
- WS-Trust
  - remote policy name, 6, 7
- WS-Trust
  - policy name, 6
  - status, 6
- WS-Trust
  - remote path, 6
- WS-Trust
  - enable privileged access, 7
- WS-Trust
  - remote path, 7
- WS-Trust
  - saving WS-Trust server configuration settings, 8
- WS-Trust
  - adding run-time WS-Trust policy, 8
- WS-Trust screen terms, 6