



# **Forum Sentry™ Version 9**

## **Tasks Management 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™, Sentry™ Web Services Security Gateway, Presidio™ OpenPGP Security Gateway, Forum FIA Gateway™, 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 © 2002-2025 Forum Systems, Inc. – All Rights Reserved.

Forum Sentry™ Version 9 Tasks Management Guide, published November 2025.

D-ASF-SE-670139

## Table of Contents

INTRODUCTION TO THE TASK MANAGEMENT GUIDE .....	1
Audience for the Task Management Guide .....	1
Conventions Used for the Task Management Guide.....	1
DOCUMENTS .....	2
Load a Sample Document from a File.....	2
Set a Sample Document as the Default Sample Document in the System .....	4
TASK LIST GROUPS and TASK LISTS .....	5
TASK LISTS.....	5
Add a Task List.....	5
Locking a Task List.....	6
TASK LIST GROUPS .....	6
Add a Task List Group.....	7
Add a Task List to a Task List Group .....	7
Processing All Task List in Sequence regardless of Condition .....	7
Processing Task Lists By Condition.....	7
Processing Task Lists By Hierarchy .....	8
Locking a Task List Group.....	9
SYSTEM TASK LIST GROUPS .....	9
TASKS ON THE SYSTEM.....	10
CONDITIONAL IDENTIFICATION TASKS.....	11
TASK: IDENTIFY DOCUMENT .....	11
Identify Document Screen Terms .....	11
MEDIATION AND TRANSFORMATION TASKS .....	14
TASK: Add XML NODE .....	14
Add XML Node Task Screen Terms .....	14
TASK: CONVERT COPYBOOK.....	14
Convert CopyBook Task Screen Terms .....	15
TASK: CONVERT CSV .....	15
Convert CSV Task Screen Terms.....	16
TASK: CONVERT JSON .....	16
Convert JSON Task Screen Terms .....	16
TASK: CONVERT SOAP.....	17
Convert SOAP To XML Task Screen Terms .....	17
TASK: Convert Value .....	18
Convert Value Task Screen Terms.....	18
TASK: ENRICH MESSAGE.....	19
Enrich Message Screen Terms.....	20
TASK: EXECUTE JAVASCRIPT .....	21
Execute JavaScript Task Screen Terms.....	21
Passing JavaScript Variables Into the JavaScript Code.....	22
TASK: PROCESS ATTACHMENTS .....	23
Process Attachments Task Screen Terms.....	24
TASK: REMOVE TRANSPORT HEADER .....	24
Replace Remove Transport Header Task Screen Terms.....	25
TASK: REMOVE WS-SECURITY HEADER .....	25
Remove WS-Security Header Task Screen Terms .....	25
Options Available When Removing a WS-Security Header .....	25
TASK: REMOVE XML NODE.....	26
Remove XML Node Task Screen Terms .....	26
TASK: REPLACE DOCUMENT .....	26
Replace Document Task Screen Terms .....	26
TASK: TRANSFORM DOCUMENT (XSLT) .....	27
Tranform Document Task Screen Terms.....	27

ATTRIBUTE MAPPING TASKS .....	28
TASK: MAP ATTRIBUTES AND HEADERS .....	28
Map Attributes and Headers Screen Terms.....	28
TASK: MAP ATTRIBUTES FROM XML .....	30
Map Attributes from XML Task Screen Terms.....	30
TASK: MAP ATTRIBUTES TO XML .....	31
Map Attributes to XML Task Screen Terms .....	32
TASK: MAPPING TABLE .....	33
Mapping Table Task Screen Terms .....	33
TASK: QUERY DATABASE .....	34
Query Data Source Task Screen Terms.....	34
Behavior .....	35
Example 1: IN Clause Expansion .....	35
Example 2: OR Condition Handling .....	35
TASK: QUERY LDAP .....	37
Query LDAP Task Screen Terms .....	37
USER IDENTITY AND ACCESS CONTROL TASKS .....	38
TASK: IP ACL .....	38
IP ACL Task Screen Terms .....	38
TASK: LOGOUT .....	38
Logout Task Screen Terms.....	39
TASK: USER IDENTITY AND ACCESS CONTROL .....	39
Access Control Lists.....	40
Access Control Options with User Identity and Access Control Tasks.....	40
Prerequisites for All User Identity and Access Control Tasks.....	41
User Identity and Access Control Task Screen Terms .....	41
Protocol-based User Identity and Access Control .....	42
Add User Identity and Access Control by XML Mapping Task .....	43
Add User Identity and Access Control by Digital Signature Task .....	43
FLOW CONTROL TASKS .....	43
TASK: ABORT PROCESSING.....	44
Abort Task Screen Terms .....	44
TASK: CACHE RESPONSE.....	44
Cache Response Task Screen Terms .....	44
TASK: DELAY PROCESSING .....	45
Delay Processing Screen Terms .....	45
TASK: FOR LOOP .....	45
For Loop Screen Terms .....	46
Configuration Parameters .....	46
Iteration Logic.....	47
Break Condition Functionality .....	47
Async vs. Sync Modes (Break Behavior).....	47
Example Scenario (Break Condition) .....	48
TASK: REDIRECT .....	49
Redirect Example .....	49
Redirect Screen Terms .....	49
TASK: REMOTE ROUTING .....	50
Content-based Routing Using the Remote Routing Task.....	50
Remote Routing Screen Terms .....	50
TASK: WS-ADDRESSING .....	51
WS-Addressing Task Screen Terms.....	52
VALIDATION AND CONFORMANCE TASKS .....	53
TASK: VALIDATE DOCUMENT STRUCTURE (Schema Validation) .....	53
Validate Document Structure Task Screen Terms .....	54
Overview of Validating with a Standalone or Compound Schema .....	54
TASK: VALIDATE JSON .....	55

Validate JSON Task Screen Terms .....	55
TASK: VALIDATE X509 CERTIFICATES .....	55
Validate X509 Certificates Task Screen Terms .....	56
LOGGING AND ARCHIVING TASKS.....	57
TASK: ALERT TASK .....	57
Alert Task Screen Terms .....	58
TASK: ARCHIVE DOCUMENT .....	59
Archive Document Task Screen Terms .....	59
TASK: DISPLAY WSDL URIs.....	60
Display WSDL URIs Task Screen Terms .....	60
TASK: LOG .....	60
Log Message Screen Terms .....	61
TASK: LOG TRANSACTION PROPERTIES .....	61
Mapping Table Task Screen Terms .....	61
CREDENTIAL GENERATION TASKS .....	62
TASK: GENERATE PASSWORD .....	62
Mapping Table Task Screen Terms .....	62
TASK: SAML ASSERTION.....	63
SAML Assertion Task Terms .....	64
TASK- WS-SECURITY HEADER.....	66
Prerequisites for All WS-Security Header Tasks .....	68
Replay Verification with WS-Security Header Tasks .....	68
WS-Security Header Task Wizard Terms .....	68
SECURITY PROCESSING TASKS.....	70
TASK: DECRYPT ELEMENTS .....	70
Element-Level and Content-Level Decryption .....	70
Decryption Screen Terms .....	70
TASK: ENCRYPT ELEMENTS .....	72
Encryption Screen Terms.....	74
TASK: JWE ENCRYPTION .....	75
JWE Encryption Screen Terms.....	75
Encryption Policy and JWE Algorithms.....	77
Input and Output Format Details .....	77
Attribute Name and JSON Key .....	77
Signature Policy .....	77
Example Configuration.....	77
TASK: JWE DECRYPTION .....	79
JWE Decryption Screen Terms.....	79
Document Handling.....	80
TASK: JWS SIGNATURE.....	81
JWS Signature Screen Terms .....	81
Input and Output Format Details:.....	82
Attribute Name and JSON Key: .....	82
TASK: JWS VERIFICATION .....	83
JWS Verification Screen Terms .....	83
Document Handling.....	84
TASK: PATTERN MATCH.....	84
Pattern Match Task Screen Terms .....	84
PATTERN MATCH POLICIES.....	85
TASK: RECEIVE SIGNATURE CONFIRMATION .....	87
Receive Signature ConfirmationTask Screen Terms.....	87
TASK: SEND SIGNATURE CONFIRMATION .....	87
Send Signature ConfirmationTask Screen Terms .....	88
TASK: SIGN DOCUMENT.....	88
Signature Types Supported .....	88
Key Types and Profiles Supported .....	88

Signature Task Screen Terms .....	88
Canonicalizing XML Signatures .....	90
Signature Transform Definitions .....	91
Filter Embedded Content Signatures Checkbox Definitions .....	92
Apply ebXML Signatures with SOAP Attachments .....	92
TASK: VERIFY DOCUMENT SIGNATURE .....	92
Signature Types Supported for Verification .....	92
Verify ebXML Signatures .....	93
Verify Attachments .....	94
Option Available For Removing a Signature .....	94
Verify Signature with Allow XPath and XSLT Transforms Option .....	94
Option Available Requiring Signatures on All Attachments .....	94
Verify Document Signature Task Screen Terms .....	95
TASK: VIRUS SCAN .....	96
Virus Scan Task Screen Terms .....	96
TASK: ZIP CONTENTS PROCESSING .....	98
Virus Scan Task Screen Terms .....	98
ZIP Content Processing Request Filter Requirement .....	99
APPENDIX .....	100
Appendix A - Constraints in Tasks Management Guide .....	100
Appendix B - Encrypt Screen Reference Chart in Tasks Management Guide .....	101
Appendix C - Signature Screen Reference Chart in Tasks Management Guide .....	102
Appendix D - Example Compound Schema Reference Chart in Tasks Management Guide .....	103
INDEX .....	104

## List of Figures

Figure 1: Options Available in the Encrypt Screen. ....	101
Figure 2: Options Available in the Signature Screen. ....	102
Figure 3: Example Compound Schema. ....	103

# INTRODUCTION TO THE TASK MANAGEMENT GUIDE

## Audience for the Task Management Guide

The *Forum Systems Sentry™ Version 9 Tasks Management Guide* defines the comprehensive set of document processing rules that can be created to map, transform, identify, and otherwise manipulate the transaction. The list of tasks includes:

- Abort Processing
- Archive Document
- Convert JSON
- Convert SOAP to XML
- Convert XML to SOAP
- Convert XML Node
- Delay Processing
- Decrypt Elements
- Display WSDLs URIs
- Encrypt Elements
- Enrich Message
- Identify Document
- Logout
- Log
- Map Attributes to XML
- Map Attributes from XML
- Map Attributes and Headers
- Pattern Match
- Query Data Source
- Receive Signature Confirmation
- Replace Document
- Remote Routing
- Remove WS-Security Header
- Remove XML Node
- SAML Assertion
- Send Signature Confirmation
- Sign Document
- Transform Document
- User Identity & Access Control
- Validate Document Structure
- Validate JSON
- Validate X.509 Certificates
- Verify Document Signature
- Virus Scan
- WS-Security Header
- WS-Addressing
- WS Secure Conversation
- XKMS Service

## Conventions Used for the Task Management Guide

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: **\*\*\*\*\***

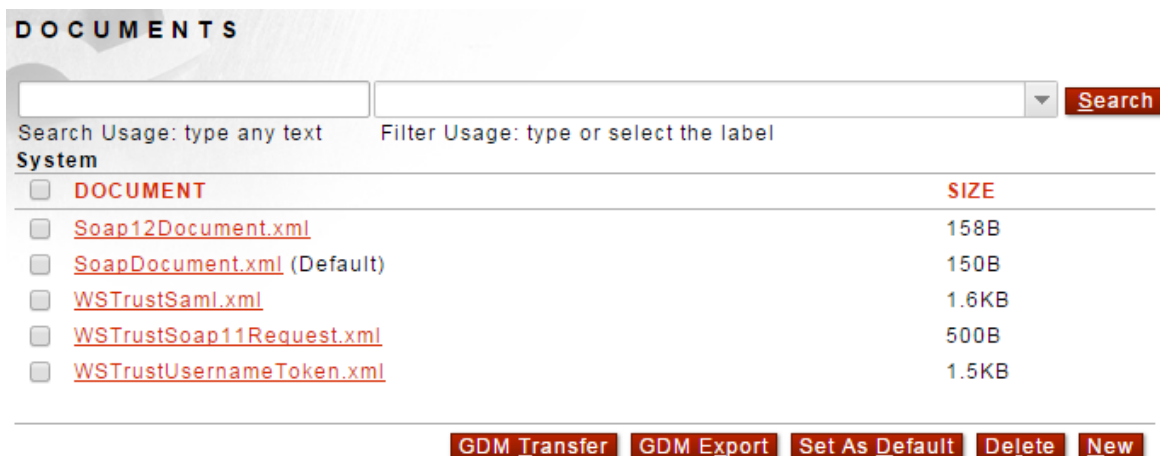
Customers with plug-in licenses on the system, confirm that your Integration License is visible on the General Info screen under the SUPPORTED FEATURES section.

Any specifications and constraints referenced in this volume appear in the Appendices of this volume.

## DOCUMENTS

The DOCUMENTS screen provides a method for creating or loading sample documents to use within Tasks in order to more quickly isolate specific document types to operate on, or specific parts of the document to act upon for tasks such as signing, encryption, mapping, transforming, removing, etc.

The DOCUMENTS screen on the Navigator displays a collection of all sample files currently in the system.



**DOCUMENTS**

Search Usage: type any text      Filter Usage: type or select the label

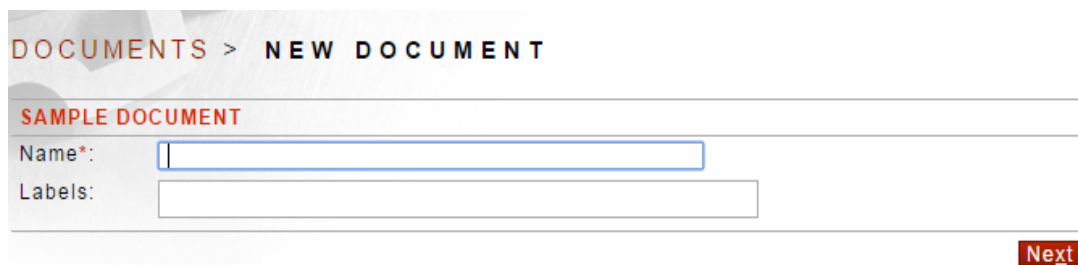
**System**

<input type="checkbox"/> DOCUMENT	SIZE
<input type="checkbox"/> <a href="#">Soap12Document.xml</a>	158B
<input type="checkbox"/> <a href="#">SoapDocument.xml</a> (Default)	150B
<input type="checkbox"/> <a href="#">WSTrustSaml.xml</a>	1.6KB
<input type="checkbox"/> <a href="#">WSTrustSoap11Request.xml</a>	500B
<input type="checkbox"/> <a href="#">WSTrustUsernameToken.xml</a>	1.5KB

[GDM Transfer](#) [GDM Export](#) [Set As Default](#) [Delete](#) [New](#)

### Load a Sample Document from a File

Follow these steps to load a sample document from a file:



**DOCUMENTS > NEW DOCUMENT**

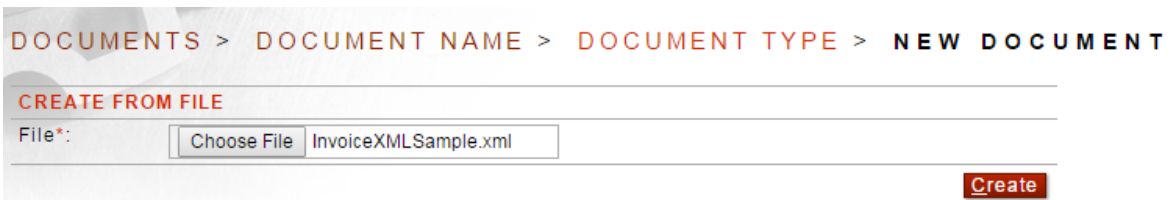
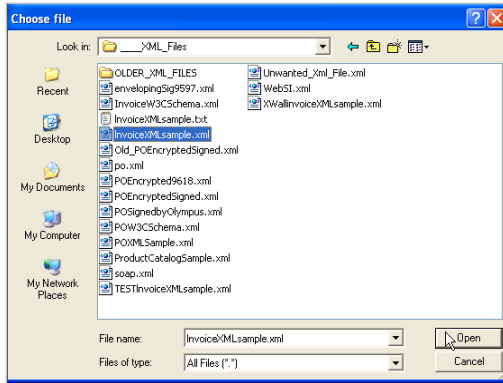
**SAMPLE DOCUMENT**

Name\*:

Labels:

[Next](#)





1. From the **RESOURCES** section of the Navigator, select **Documents** and the DOCUMENTS screen appears.
2. Select **New**, and the NEW DOCUMENT screen appears.
3. Select the **File** radio button, and then click **Browse**. The Choose file screen appears.
4. Navigate to and highlight a **file**. The filename populated the File name field.
5. Select **Open** and the NEW DOCUMENT screen refreshes.
6. Select **Save** and the **DOCUMENTS** screen refreshes.

## View a Sample Document

To view a sample document, go to the **RESOURCES-> Documents** menu and click on the document name hyperlink shown.

## Set a Sample Document as the Default Sample Document in the System

Follow these steps to set a sample document as the system default sample document:

**DOCUMENTS**

Search Usage: type any text      Filter Usage: type or select the label

**No Labels**

<input type="checkbox"/> DOCUMENT	SIZE
<input checked="" type="checkbox"/> <a href="#">InvoiceXMLSample.xml</a>	4B

**System**

<input type="checkbox"/> DOCUMENT	SIZE
<input type="checkbox"/> <a href="#">Soap12Document.xml</a>	158B
<input type="checkbox"/> <a href="#">SoapDocument.xml</a> (Default)	150B
<input type="checkbox"/> <a href="#">WSTrustSaml.xml</a>	1.6KB
<input type="checkbox"/> <a href="#">WSTrustSoap11Request.xml</a>	500B
<input type="checkbox"/> <a href="#">WSTrustUsernameToken.xml</a>	1.5KB

- From the **RESOURCES** section of the Navigator, select **Documents** and the DOCUMENTS screen appears.
- Check the **checkbox** prefacing the sample document to be designated as the system default sample document, and then select **Set As Default**.
- The DOCUMENTS screen refreshes.

## TASK LIST GROUPS and TASK LISTS

Task List Groups and Task Lists are the workflow policies that enable the ability to apply task processing to the request and/or the responses of Content Policies. Task List Groups contain one or more Task List to provide more complex if...then type processing logic.

### TASK LISTS

A Task List is a grouping of sequentially ordered Tasks which create a workflow for transaction processing. This workflow can be applied to the request and/or the response. Task Lists are located under the **Gateway->Task Policies->Task Lists** menu. Task Lists are later consumed by Task List Groups, or Content policies that accept messages into the system.

### Add a Task List

To Add a Task List, go to **Gateway->Task Policies->Task Lists** and click on the New button.

### Run the Task List

At design time the Administrator can see the results of the Tasks by click on the **Run** button. This will use the provided credentials and sample document to run each Task in the Task List in sequence and show the resulting document in a popup window.

### Set Design-time Task Validation

Each Task item created will show the results of the previous task items that are sequentially before it. This is to ensure that the task will see the result of the task manipulations up to that point (such as a new DSIG added, or a transformation change, etc). Sometimes the sample document does not match the task list which will result in a design-time error. To work around this error, click on the **Settings** button and click on "Ignore Sample Document Errors" checkbox.

TASK LISTS > TASK LIST: TASK\_LIST\_FOR\_INVOICES >  
SETTINGS

USER CREDENTIALS

User Name: walker \*

Password: \*\*\*\*\* \*

Resource: forum \*

Certificate: [None] ▼

SETTINGS

☒ Ignore sample document errors

Next

### Promote or Demote Tasks

To raise or lower the task item sequence in the list, simply click on the task item and drag it up or down. The Task # sequence will update accordingly.

## Task Item Sequence Order in a Task List

Tasks are performed in sequential order as they appear from the top down in the WebAdmin interface. Tasks can be moved up or down in the sequence simply by clicking on the task item and dragging the item above or below. The numbers that appear under the # symbol indicate the task sequence (1,2,3,...)

### Tasks

To add a new task, click the New button. To insert a task at a specified location, check the box for the existing task, then click New. Tasks can be reordered by Drag and Drop and are automatically saved.

<input type="checkbox"/>	#	TASK NAME	TASK TYPE	STATUS
<input type="checkbox"/>	1	<a href="#">Delay Processing</a>	Delay Processing	<span style="color: green;">●</span>
<input type="checkbox"/>	2	<a href="#">Map Attributes and Headers</a>	Map Attributes and Headers	<span style="color: green;">●</span>
<input type="checkbox"/>	3	<a href="#">Replace Document</a>	Replace Document	<span style="color: green;">●</span>

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

## Locking a Task List

The lock checkbox will ensure that the Task List is not modified by any incoming FSG configuration import.

## TASK LIST GROUPS

A Task List Group is a collective representation of one or more Task Lists. The Task Lists in a Task List Group are later consumed by Content policies for workflow processing of the messages. Task List Groups are reusable, so naming them with an easily-recognizable name is advisable.

### TASK LIST GROUP

▼

[Search](#)

Search Usage: type any text
Filter Usage: type or select the label

[Always Show Expanded](#)

**System**

<input type="checkbox"/>	NAME		ASSOCIATIONS
<input type="checkbox"/>	<a href="#">System Request Group</a> (0)		<a href="#">Policies</a> (0)
<input type="checkbox"/>	<a href="#">System Response Group</a> (0)		<a href="#">Policies</a> (0)

**Credential Generation**

<input type="checkbox"/>	NAME		ASSOCIATIONS
<input type="checkbox"/>	<a href="#">SAML Assert Generation</a> (1)		<a href="#">Policies</a> (0)

**Security Processing**

<input type="checkbox"/>	NAME		ASSOCIATIONS
<input type="checkbox"/>	<a href="#">Encrypt_Sign</a> (1)		<a href="#">Policies</a> (0)
<input type="checkbox"/>	<a href="#">Encrypt_Sign-2</a> (1)		<a href="#">Policies</a> (0)
<input type="checkbox"/>	<a href="#">UserID Att sessionnum map XML</a> (1)		<a href="#">Policies</a> (0)

#	TASK LIST	STATUS
1	<a href="#">UserID Att sessionnum map XML</a>	<span style="color: green;">●</span>

**No Labels**

<input type="checkbox"/>	NAME		ASSOCIATIONS
<input type="checkbox"/>	<a href="#">IFA XML TLG</a> (1)		<a href="#">Policies</a> (1)
<input type="checkbox"/>	<a href="#">Task List For Invoices</a> (1)		<a href="#">Policies</a> (0)

[GDM Transfer](#)
[GDM Export](#)
[Delete](#)
[New](#)

The sequence for populating a Task List Group with one or more Task Lists is:

- Create a Task List Group from the Task List Group screen.
- On the TASK LIST GROUP DETAILS screen, add one or more Task Lists.

- On a Content policy, apply the Task List Group to the request or the responses.

## Add a Task List Group

To add a Task List Group, click on the New button under Task List Groups.

TASK LIST GROUP > TASK LIST GROUP DETAILS

TASK LIST GROUP DETAILS

Task List Group Name\*:

Task\_List\_Group

Description:

Process Each Task List Below in Sequence:

☐

Lock:

☐

Labels:

TASK LIST

Type or select label name

Type or select item name

Add

Enable

Disable

Remove

Apply

Save

## Add a Task List to a Task List Group

Users may add one or more Task Lists to a Task List Group. When adding Task Lists you can use the Label filter to filter only those Task Lists with that Label to reduce the number of Task Lists to select from.

There are a few considerations when adding multiple Task Lists to a Task List Group

### Processing All Task List in Sequence regardless of Condition

If you want to process all of the Task Lists in succession, check this box. This results in the same behavior as adding all of the Task items to one single Task List.

### Processing Task Lists By Condition

The more Common Task List approach is to add multiple Task Lists where each Task List starts with an Identity Document task. The Identity Document task in each Task List must be the first Task Item in the list. Then when the workflow process begins and calls the Task List Group, the proper Task List can be invoked based on the Identity Document task within each associated Task List. If there are multiple Task Lists where the Identity Document Task matches, the first in hierarchy will be the only one which runs.

## Processing Task Lists By Hierarchy

When adding Task Lists, you will be presented by the ability to move the Task List up, down, right, and left depending on the hierarchy.

### TASK LIST GROUP > TASK LIST GROUP DETAILS

#### TASK LIST GROUP DETAILS

Task List Group Name*:	<input type="text" value="Sample Group 1"/>
Description:	<input type="text"/>
Process Each Task List Below in Sequence:	<input type="checkbox"/>
Lock:	<input type="checkbox"/>
Labels:	<input type="text"/>

#### TASK LIST

Sample 2 <a href="#">Edit</a>	
ReplacePingResponse <a href="#">Edit</a>	
Sample 2 <a href="#">Edit</a>	

<input type="text" value="Type or select label name"/>	<input type="text" value="Type or select item name"/>	<input type="button" value="Add"/>
<div><input type="button" value="Enable"/> <input type="button" value="Disable"/> <input type="button" value="Remove"/> <input type="button" value="Apply"/> <input type="button" value="Save"/></div>		

For example, if you want to make a Task List a child of another Task List such that the parent must be processed first, you can use the Right arrow to indent the Task List as a child. In the example below, the ReplacePingResponse task is a child of the Sample 2 task. The “First Match” indicator means that only the child task with a matching Identity Document task will be run. If a Task List is encountered without an Identity Document task, it will be run by default.

#### TASK LIST

▼ Sample 2 <a href="#">Edit</a>	First Match	
ReplacePingResponse <a href="#">Edit</a>		
Sample 2 <a href="#">Edit</a>		

If the “Process All” option is selected, all child Task Lists at that hierarchy will be run, as per the example below.

---

**TASK LIST** 

▼ Sample 2 [Edit](#)

Process All 

ReplacePingResponse [Edit](#)



### Locking a Task List Group

The lock checkbox will ensure that the Task List Group is not modified by any incoming FSG configuration import.

### SYSTEM TASK LIST GROUPS

System Task List Groups are policies where tasks can be associated that will apply to every transaction across every policy on the system. There is a Request and a Response System Task List Group.

## TASKS ON THE SYSTEM

Tasks are used to perform processing actions for different triggered transactions based on identification criteria, or simply by association of a task to a task list group. All tasks are created within Task Lists and then these Task Lists are associated with a Task List Group. A Task List Group is the policy object that is then associated with the transaction policies such as WSDL, XML, and HTML in order to accomplish the set of processing tasks for the transactions coming to that policy.

The tasks discussed in this section include:

### TASK LISTS > TASK LIST: SAMPLE 2 > NEW TASK

TASK TYPE		
<b>Conditional Identification</b> <ul style="list-style-type: none"><li><input type="radio"/> Identify Document</li></ul>	<b>User Identity and Access Control</b> <ul style="list-style-type: none"><li><input type="radio"/> IP ACL</li><li><input type="radio"/> Logout</li><li><input type="radio"/> User Identity &amp; Access Control</li></ul>	<b>Credential Generation</b> <ul style="list-style-type: none"><li><input type="radio"/> Generate Password</li><li><input type="radio"/> SAML Assertion</li><li><input type="radio"/> WS-Security Header</li></ul>
<b>Mediation and Transformation</b> <ul style="list-style-type: none"><li><input type="radio"/> Add XML Node</li><li><input type="radio"/> AS2</li><li><input type="radio"/> Convert Copybook</li><li><input type="radio"/> Convert CSV</li><li><input type="radio"/> Convert JSON</li><li><input type="radio"/> Convert SOAP</li><li><input type="radio"/> Convert Value</li><li><input type="radio"/> ebMS</li><li><input type="radio"/> Enrich Message</li><li><input type="radio"/> Execute JavaScript</li><li><input type="radio"/> Process Attachments</li><li><input type="radio"/> Remove Transport Header</li><li><input type="radio"/> Remove WS-Security Header</li><li><input type="radio"/> Remove XML Node</li><li><input type="radio"/> Replace Document</li><li><input type="radio"/> Transform Document</li></ul>	<b>Flow Control</b> <ul style="list-style-type: none"><li><input checked="" type="radio"/> Abort Processing</li><li><input type="radio"/> Cache Response</li><li><input type="radio"/> Delay Processing</li><li><input type="radio"/> For Loop</li><li><input type="radio"/> Redirect</li><li><input type="radio"/> Remote Routing</li><li><input type="radio"/> WS-Addressing</li></ul> <b>Validation and Conformance</b> <ul style="list-style-type: none"><li><input type="radio"/> Validate Document Structure</li><li><input type="radio"/> Validate JSON</li><li><input type="radio"/> Validate X.509 Certificates</li></ul> <b>Logging and Archiving</b> <ul style="list-style-type: none"><li><input type="radio"/> Alert</li><li><input type="radio"/> Archive Document</li><li><input type="radio"/> Display WSDLs URIs</li><li><input type="radio"/> Log</li><li><input type="radio"/> Log Transaction Properties</li></ul>	<b>Security Processing</b> <ul style="list-style-type: none"><li><input type="radio"/> Decrypt Elements</li><li><input type="radio"/> Encrypt Elements</li><li><input type="radio"/> JWE Encryption</li><li><input type="radio"/> OpenPGP</li><li><input type="radio"/> Pattern Match</li><li><input type="radio"/> Receive Signature Confirmation</li><li><input type="radio"/> Send Signature Confirmation</li><li><input type="radio"/> Sign Document</li><li><input type="radio"/> Verify Document Signature</li><li><input type="radio"/> Virus Scan</li><li><input type="radio"/> XKMS Service</li></ul>
<b>Attribute Mapping</b> <ul style="list-style-type: none"><li><input type="radio"/> Map Attributes and Headers</li><li><input type="radio"/> Map Attributes from XML or JSON</li><li><input type="radio"/> Map Attributes to XML or JSON</li><li><input type="radio"/> Mapping Table</li><li><input type="radio"/> Query Database</li><li><input type="radio"/> Query LDAP</li></ul>		

Within each of these task types are various settings that allow a complex set of processing tasks to be deployed on the gateway to process traffic. Document processing tasks provide comprehensive coverage across industry standards. Task processing provide integration capabilities with disparate vendors systems since Sentry can consume messages in virtually any format, and convert them to virtually any other format. For example, a Digital Signature from one specification can be consumed and a new DSIG can be generated using a different specification that the other system understands. These types of actions can greatly optimize integration time and provide seamless coherence to complex architectures.



## CONDITIONAL IDENTIFICATION TASKS

In order to perform conditional logic of when to invoke a Task List Group, the First Task Item must be the Identity Document Task. This Task provides criteria to identify aspects of a request or response based on the message content, headers, etc. These rules enable the inspection of the request or response transaction information to determine if this Task List workflow is to be the one invoked to handle the processing. The Identity Document task is used heavily for Task List Group conditional processing of which Task List to invoke dynamically based on the request or response attributes.

### TASK: IDENTIFY DOCUMENT

The Identify Document task is used to designate which documents/transactions are to be identified as targets to process the specific task list. When defined, the identity document task is the first task in the list. This task is responsible for identifying which transactions match to this Task List and thus whether the Task List will be invoked to process the subsequent tasks in the Task List.

**Note:** Task Lists are not required to have an Identify Document rule defined, but when this task is not defined, the Task List will always be invoked when associated, regardless of document or message criteria.

When multiple Task Lists exist in a Task List Group, the task will trigger based on the hierarchy of the Task List in the Task List Group and based on the Identity Document matching rules per the most specific rule set defined.

When using the Identify Document task, the top section of the screen is the Header Filters for matching non-document related items, the bottom section is the Document Filters for matching document content such as XML or JSON.

### Identify Document Screen Terms

While using the Identify Document task, please consider the following terms and definitions:

TERM	DESCRIPTION OF OPTIONS
<b>IDENTIFY</b>	
Task Name	Display name for the task
<b>HEADER FILTERS</b>	
Filter Type	The source of the input to target for the comparison. Sources include: <ul style="list-style-type: none"><li>• <b>Constant</b><ul style="list-style-type: none"><li>○ A static value specified directly on the policy</li></ul></li><li>• <b>Protocol Header</b><ul style="list-style-type: none"><li>○ If the task is operating on the request, this represents the request header. If the task is operating on the response, this setting represents the response header</li></ul></li><li>• <b>Request Header</b><ul style="list-style-type: none"><li>○ The header from the inbound request from the client</li></ul></li><li>• <b>Response Header</b><ul style="list-style-type: none"><li>○ The header from the response from the back-end system</li></ul></li><li>• <b>User Attribute</b><ul style="list-style-type: none"><li>○ An attribute from LDAP, Active Directory, STS Identity Broker, Siteminder, or any other supported Sentry</li></ul></li></ul>

	<p>identity adapter that returns user attributes with the authentication response.</p> <ul style="list-style-type: none"> <li>• <b>X509 Attribute</b> <ul style="list-style-type: none"> <li>○ OID and other attributes within an X509 certificate</li> </ul> </li> <li>• <b>Query Parameter</b> <ul style="list-style-type: none"> <li>○ Each query string parameter from the inbound URI is available as an attribute to map</li> </ul> </li> <li>• <b>Cookie</b> <ul style="list-style-type: none"> <li>○ A cookie from the header</li> </ul> </li> <li>• <b>HTTP Method</b> <ul style="list-style-type: none"> <li>○ The HTTP Method</li> </ul> </li> <li>• <b>Request Path</b> <ul style="list-style-type: none"> <li>○ The Path only portion of the inbound client request URL</li> </ul> </li> <li>• <b>Request URL</b> <ul style="list-style-type: none"> <li>○ The full URL of the inbound client request</li> </ul> </li> <li>• <b>Username</b> <ul style="list-style-type: none"> <li>○ The currently authenticated client's username</li> </ul> </li> <li>• <b>Source IP Address</b> <ul style="list-style-type: none"> <li>○ The current client's source IP</li> </ul> </li> <li>• <b>HTTP Status Code</b> <ul style="list-style-type: none"> <li>○ The response code from the back-end system response</li> </ul> </li> <li>• <b>HTTP Status Message</b> <ul style="list-style-type: none"> <li>○ The response status message (associated with the response code) from the back-end system</li> </ul> </li> <li>• <b>Full Document</b> <ul style="list-style-type: none"> <li>○ The full request document if this processing is associated with the request, or the full response document if this processing is associated with the response.</li> </ul> </li> </ul>
Header Name	If the filter type selected is Protocol Header, Request Header, or Response header then this parameter is the name of the header to use for the identification target.
Comparator	The function to use to compare the source value with the target value
Value Type	<p>The target value for the identification comparison. Values Include:</p> <ul style="list-style-type: none"> <li>• <b>Constant</b> <ul style="list-style-type: none"> <li>○ A static specific value in the value field</li> </ul> </li> <li>• <b>Password</b> <ul style="list-style-type: none"> <li>○ A static specific value in the value field which is obfuscated with * characters in the GUI field</li> </ul> </li> <li>• <b>Protocol Header</b> <ul style="list-style-type: none"> <li>○ If the task is operating on the request, this represents the request header. If the task is operating on the response, this setting represents the response header</li> </ul> </li> <li>• <b>Request Header</b> <ul style="list-style-type: none"> <li>○ The header from the inbound request from the client</li> </ul> </li> <li>• <b>Response Header</b> <ul style="list-style-type: none"> <li>○ The header from the response from the back-end system</li> </ul> </li> <li>• <b>User Attribute</b> <ul style="list-style-type: none"> <li>○ A general attribute type the can be referenced by other tasks</li> </ul> </li> </ul>

- 
- **Query Parameter**
    - A target name value pair to add to the URI for the back-end server request from Sentry
  - **Cookie**
    - Using this setting will result in a SET-COOKIE header being created with the cookie value as the value specified.
  - **Template**
    - Used to split the value into segments to capture and isolate the target evaluation content based on separators.
  - **HTTP Method**
    - The HTTP Method
  - **Request Path**
    - The Path only portion of the inbound client request URL
  - **Request URL**
    - The full URI of the inbound client request
  - **Username**
    - The authenticated user's username
  - **Source IP Address**
    - The current client's source IP
  - **HTTP Status Code**
    - The response code from the back-end system response
  - **HTTP Status Message**
    - The response status message (associated with the response code) from the back-end system
- 

Value	The value as a constant or as a variable reference from one of the available defined value types
-------	--

---

#### DOCUMENT FILTERS

---

Path	This represents the XPath or JSONPath format used to target the specified location within the XML or JSON document to identify based on content and the matching function.
------	--

---

## MEDIATION AND TRANSFORMATION TASKS

The tasks in this section are used to manipulate the document or the header of the request or response.

### TASK: Add XML NODE

The Add XML Node task is used to add elements from XML documents.

**TASK LISTS > TASK LIST: SAMPLE 1 > TASK: ADD XML NODE**

---

**ADD XML NODE**

Task Type: Add XML Node

Task Name\*: Add XML Node

On Error: ☒ Log & Halt Processing ☐ Log & Continue

---

**ELEMENTS TO ADD TO**

☐ soap:Envelope

☐ soap:Body

---

**Elements to Add**

<input type="checkbox"/>	PARENT	NODE TYPE	PREFIX	NAME	NAMESPACE
No items to display					

#### Add XML Node Task Screen Terms

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
Parent	XPath expression that point to the parent node where to add the new node
Node Type	Element or Attribute
Prefix	The XML Node prefix
Name	The name of the new element or attribute
Namespace	The namespace to associate with the prefix

### TASK: CONVERT COPYBOOK

The Convert Copybook Task allows the user to import an existing copybook using the .cpy or .txt extension. When the copybook is completed in either format simply import the copybook into the task as shown in the screen shot below.

TASK LISTS > TASK LIST: SAMPLECOPYBOOKTASK-IN > TASK: CONVERT COPYBOOK

---

CONVERT COPYBOOK

Task Type:

Convert Copybook

Task Name\*:

Convert Copybook

On Error:

☒ Log & Halt Processing
 ☐ Log & Continue

---

IMPORT COPYBOOK

Filename:

SampleCopyBook.txt

Copybook:

Choose File

No file chosen

---

COPYBOOK SETTINGS

Record\*:

RCRD ▾

Mode:

☐ Convert Data into XML  
☐ Convert Data into JSON  
☒ Convert XML or JSON into Data  
☐ Create XML Template  
☐ Create JSON Template

---

## Convert CopyBook Task Screen Terms

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
Filename	The Copybook file
Record	The boundaries for the records
Mode	How to convert the values

More information on this task can be found in the Helpdesk at [How to Use Convert Copybook Task – Forum Systems Support](#)

## TASK: CONVERT CSV

This task will automatically convert CSV into XML or XML into CSV. Note that the inbound CSV must have a Content-Type of “text/csv;header=present” in order to treat the first Row of the CSV file as the header names. If your client can not send this Content-Type header, it can be set manually in the Task List above the ConvertCSV task via Map Attributes and Headers task and map Constant Value “text/csv;header=present” to Request Header target.

TASK LISTS > TASK LIST: NEW TASK LIST3 > TASK: CONVERT CSV

#### CONVERT CSV

Task Type:	Convert CSV
Task Name*:	<input type="text" value="Convert CSV"/>
On Error:	<input checked="" type="radio"/> Log & Halt Processing <input type="radio"/> Log & Continue
Mode:	<input checked="" type="radio"/> CSV to XML <input type="radio"/> XML to CSV
XML Root Element Name*:	<input type="text"/>
XML Root Element Namespace:	<input type="text"/>
XML Row Element Name*:	<input type="text"/>

#### Convert CSV Task Screen Terms

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
On Error	Halt processing, or continue based on error in the task processing for this task
Operation	CSV to XML XML to CSV
XML Root Element Name	The name to give to the root of the XML document when converting
XML Root Element Namespace	The namespace to give to the root of the XML document when converting
XML Row Element Name	The name to give to the parent of each XML node corresponding to each row in the CSV

#### TASK: CONVERT JSON

This task will automatically convert JSON into XML or XML into JSON.

TASK LISTS > TASK LIST: ARCHIVE DOCUMENT TASK > TASK: CONVERT JSON

#### CONVERT JSON

Task Type:	Convert JSON
Task Name*:	<input type="text" value="Convert JSON"/>
On Error:	<input checked="" type="radio"/> Log & Halt Processing <input type="radio"/> Log & Continue
Mode:	<input type="radio"/> JSON to XML <input checked="" type="radio"/> XML to JSON
<input checked="" type="checkbox"/> Remove XML Root Element Before Converting to JSON	
<input checked="" type="checkbox"/> Preserve attributes and namespaces	

#### Convert JSON Task Screen Terms

TERM	DESCRIPTION OF OPTIONS

Task Name	The name of the task
On Error	Halt processing, or continue based on error in the task processing for this task
Operation	JSON to XML XML to JSON
Remove XML Root	When checked, removes the XML root if necessary before converting to JSON format

### TASK: CONVERT SOAP

This task will automatically convert SOAP into XML, XML to SOAP, MTOM to SOAP, and SOAP to MTOM.

**TASK LISTS > TASK LIST: NEW TASK LIST3 > TASK: CONVERT SOAP**

---

**CONVERT SOAP**

Task Type:	Convert SOAP
Task Name*:	<input type="text" value="Convert SOAP"/>
On Error:	<input checked="" type="radio"/> Log & Halt Processing <input type="radio"/> Log & Continue
Operation:	<input checked="" type="radio"/> Convert SOAP to XML <input type="radio"/> Convert XML to SOAP <input type="radio"/> Convert MTOM to Soap <input type="radio"/> Convert SOAP to MTOM

### Convert SOAP To XML Task Screen Terms

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
On Error	Halt processing, or continue based on error in the task processing for this task
Operation	Convert SOAP to XML Convert XML to SOAP Convert MTOM to SOAP Convert SOAP to MTOM

## TASK: Convert Value

This task will convert, hash, encode and decode values. Options include BASE64, URL, Encrypt, Decrypt, Digest, and SHA/AES/SHA-256 hashing.

**TASK LISTS > TASK LIST: ARCHIVE DOCUMENT TASK > TASK: CONVERT VALUE**

---

**CONVERT VALUE**

Task Type: Convert Value

Task Name\*:

On Error: ☒ Log & Halt Processing ☐ Log & Continue

Operation:

<input type="radio"/> Base64 Encode	<input checked="" type="radio"/> Base64 Decode
<input type="radio"/> Hex Binary Encode	<input type="radio"/> Hex Binary Decode
<input type="radio"/> URL Encode	<input type="radio"/> URL Decode
<input type="radio"/> Encrypt	<input type="radio"/> Decrypt
<input type="radio"/> Digest	<input type="radio"/> Sign
<input type="radio"/> Uppercase	<input type="radio"/> Lowercase
<input type="radio"/> Split	<input type="radio"/> Aggregate
<input type="radio"/> Parent	<input type="radio"/> SHA/AES/SHA-256

Encryption Policy:  [Edit](#)

Decryption Policy:

Digest Algorithm:

Signature Policy:  [Edit](#)

Encoder:

☐ Convert multiple delimited values

Delimiter:

**Attributes to Convert**

#	ATTRIBUTE TYPE	ATTRIBUTE NAME	STATUS
No items to display			

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

**SELECT ELEMENTS TO CONVERT**

☐ soap:Envelope

☒ soap:Body

**Elements to Convert**

<input type="checkbox"/> ELEMENT
No items to display

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

## Convert Value Task Screen Terms

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
On Error	Halt processing, or continue based on error in the task processing for this task



Operation	Base64 Encode Base64 Decode URL Encode URL Decode Encrypt Decrypt Digest SHA/AES/SHA-256
Encryption Algorithm	Enabled when Encrypt or Decrypt is selected in the Operation. Options include AES256, AES192, AES128, and 3DES
Symmetric Key	Enabled when Encrypt or Decrypt is selected in the Operation. Value is used as the Symmetric key for the crypto operation.
Digest Algorithm	Enabled when Digest is selected in the Operation. Options for digest hashing include SHA1,SHA224,SHA384,SHA256,SHA512,RIPEMD160

Convert Value target options include:

- **Protocol Header**
  - If the task is operating on the request, this represents the request header. If the task is operating on the response, this setting represents the response header
- **Request Header**
  - The header from the inbound request from the client
- **Response Header**
  - The header from the response from the back-end system
- **User Attribute**
  - A general attribute type that can be referenced by other tasks
- **Query Parameter**
  - A target name value pair to add to the URI for the back-end server request from Sentry
- **Cookie**
  - Using this setting will result in a SET-COOKIE header being created with the cookie value as the value specified.
- **HTTP Method**
  - GET or POST
- **Request Path**
  - The request path of the inbound client request, not including query parameters
- **HTTP Status Code**
  - The response code from the back-end system response back to Sentry

## TASK: ENRICH MESSAGE

This task enables 3<sup>rd</sup> party integration, or loopback policy to another Sentry policy, to be used to extend, modify, record, transform, notify, or any other type of event or integration activity associated with enrichment of the current transaction policy and current message.

The Enrich Message task uses the existing Sentry network policy infrastructure allowing the selection of any type of remote network policy (HTTP, FTP, MQ, EMS, etc) as the end-point location for the message enrichment activity.

The Enrich Message task holds the current transaction and takes the current message and performs the following sequence:


- 1) Hold the current transaction request or response event
- 2) Determine whether to propagate headers from the request or response to the remote policy location
- 3) If a Request Task List Group is defined, a copy of the current document will be first processed against this Task Group
- 4) Sends the current transaction document to the Remote Policy and Path
- 5) Receives the response from the Remote Policy and Path
- 6) If a Response Task List Group is defined, this Task Group will be run against the response that was received. Since the message response is only used for processing against the Task Group, this is the time to use Mapping Tasks to obtain information from the 3<sup>rd</sup> party enrichment service to map back to the original document being held from step 1.

**TASK LISTS > TASK LIST: ENRICH MESSAGE > TASK: ENRICH MESSAGE**

---

**ENRICH MESSAGE**

Task Type: Enrich Message

Task Name\*:  

On Error: ☒ Log & Halt Processing ☐ Log & Continue

Error Template:  ▼

---

Remote Policy\*:  [Edit](#)

Remote Path\*:

Remote URI:

Propagate Headers: ☐

Request Task List Group:   ▼  ▼

Response Task List Group:   ▼  ▼

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

### Enrich Message Screen Terms

While using the Enrich Message task, please consider the following terms and definitions:

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
Remote Policy	The remote policy to use to communicate with the Enrichment Service
Remote Path	The remote path to use when communicating with the Enrichment Service via HTTP(S)
Propagate Headers	Indicates whether to map the inbound connection headers to the headers used in the new outbound request to the Enrichment Service
Request Task List Group	The Task Group used to process the request prior to sending to the Enrichment Service. Examples of use are transforming the request, mapping information, added authentication criteria for 3 <sup>rd</sup> party authentication, etc.
Response Task List Group	The Task Group used to process the response coming back from the Enrichment Service. It is important to note that the response document itself will be discarded after processing this Task group. If you want to map values that came back from the Enrichment Service, be sure to add Mapping Tasks to the Response Task List Group to preserve values

from this response.

## TASK: EXECUTE JAVASCRIPT

The Execute JavaScript task is used to directly invoke JavaScript code as a part of the task workflow. The JavaScript code can not contain any references to external libraries and it must be native Nashorn ECMAScript 5.1 compliant or Sentry will not be able to run the code.

The usage of the JavaScript task allows the request or response document, or stored attributes to be processed and manipulated using raw JavaScript code. This is helpful for more complex string manipulations or complex logic that may be easier to implement directly in JavaScript code functions.

Note: The Execute JavaScript task requires a special license feature for it to appear as an option in the Task List.

TASK LISTS > TASK LIST: NEW TASK LIST > TASK: EXECUTE JAVASCRIPT

EXECUTE JAVASCRIPT

Task Type:

Execute JavaScript

Task Name\*:

On Error:

☒ Log & Halt Processing ☐ Log & Continue

Filename:

Script\*:

No file selected.

MAPPING CONFIGURATION

If Output is not JSON:

☒ Map to a Single Attribute  
Attribute Name\*:   
☐ Map to a Full Document

If Output is JSON:

☐ Map to a Single Attribute  
Attribute Name:   
☒ Map to Multiple Attributes (Based on JSON Keys)  
☐ Map to a Full Document

JSON Handling:

☐ Convert to String  
☐ Convert to JSON  
☒ Leave as is

### Execute JavaScript Task Screen Terms

While using the Process Attachments task, please consider the following terms and definitions. The output of the JavaScript code will be evaluated to determine if the format is in JSON format or non-JSON (i.e. a text string) format. The Mapping Configuration options provide various ways to take the output of the JavaScript code and map back to the Sentry document (request or response) or user attribute.

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
Script	The actual contents of Nashorn ECMAScript 5.1 compliant JavaScript code

If Output is not JSON	<ul style="list-style-type: none"> <li>• Map to a Single Attribute. <ul style="list-style-type: none"> <li>○ Maps to an attribute variable specified in the Attribute Name field.</li> </ul> </li> <li>• Map to a Full Document. <ul style="list-style-type: none"> <li>○ Maps to the current full document in the location of the task list where the JavaScript code is being run.</li> </ul> </li> </ul>
If Output is JSON	<ul style="list-style-type: none"> <li>• Map to a Single Attribute. <ul style="list-style-type: none"> <li>○ Maps to an attribute variable specified in the Attribute Name field.</li> </ul> </li> <li>• Map to Multiple Attributes (Based on JSON Keys). <ul style="list-style-type: none"> <li>○ Maps to a set of attribute variables specified in the JSON output values.</li> </ul> </li> <li>• Map to a Full Document. <ul style="list-style-type: none"> <li>○ Maps to the current full document in the location of the task list where the JavaScript code is being run.</li> </ul> </li> </ul>
JSON Handling	<ul style="list-style-type: none"> <li>• Convert to String. <ul style="list-style-type: none"> <li>○ Treat the JSON output from the JavaScript code as a string value.</li> </ul> </li> <li>•  <ul style="list-style-type: none"> <li>○ Convert to JSON. Attempt to convert the output from the JavaScript code as a properly formatted JSON message</li> </ul> </li> <li>• Leave as IS. <ul style="list-style-type: none"> <li>○ Don't alter the output of the JavaScript code.</li> </ul> </li> </ul>

### Passing JavaScript Variables Into the JavaScript Code

The User Attributes that are configured at the point in the task list where you are running the Execute JavaScript task are all available for usage in the JavaScript code. To use the variables in JavaScript code, simply use the variable name as the name as the attribute.

#### Example 1 – Map a String Value from JavaScript Output to a User Attribute

Task 1: Map Attributes and Headers. Map the inbound document to an attribute named **payload**

Task 2: Execute JavaScript. If Output is not JSON, Map to a single Attribute. Use the attribute name **payloadLength**.

The variable **payload** will be automatically created in JavaScript and can be referenced directly in the JavaScript code such as:

```
function countCharacters(str) {
  if (str == null) {
    return 0;
  }
}
```

```

    }
    return String(str).length;
}

countCharacters(payload);

```

This will return a string value of the character count in the variable payload and because the option was “Map to a single Attribute” the value will be stored in a user attribute named **payloadLength**.

## Example 2 - Map Multiple JSON Values from JavaScript Output to User Attributes

Task 1: Execute JavaScript. If Output is JSON, Map to Multiple Attributes (Based on JSON Keys).

```

function getCurrentDateTime() {
    var d = new Date();
    // ISO string like "2025-11-11T15:04:05.123Z"
    var iso = d.toISOString();
    return {
        date: iso.slice(0, 10),          // "YYYY-MM-DD"
        time: iso.slice(11, 19),        // "HH:MM:SS" (UTC)
        iso: iso,                       // full ISO in UTC
        epochMillis: d.getTime(),       // handy for sorting
        tzOffsetMinutes: -d.getTimezoneOffset()
    };
}

getCurrentDateTime();

```

This will return a JSON message:

```

{
  "date": "2025-11-11",
  "time": "21:23:37",
  "iso": "2025-11-11T21:23:37.674Z",
  "epochMillis": 1.762896217674E12,
  "tzOffsetMinutes": -300
}

```

The example provided “Map to Multiple Attributes (Based on JSON Keys)” option will automatically map to user attributes named **date**, **time**, **iso**, **epochMillis**, and **tzOffsetMinutes**.

## TASK: PROCESS ATTACHMENTS

The Process Attachments task is used to match attachments by content-type or other attachment header criteria in order to determine the operation to perform. Operations include:

- Remove
- Block
- Base64 Encode

## TASK LISTS > TASK LIST: NEW TASK LIST2 > TASK: PROCESS ATTACHMENTS

### PROCESS ATTACHMENTS

Task Type: Process Attachments

Task Name\*: Process Attachments

Match Header: ☒

Header Name\*:

Comparator:

=

Header Value:

Operation:

Remove

### Process Attachments Task Screen Terms

While using the Process Attachments task, please consider the following terms and definitions:

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
Match Header	Matches the header of the attachment section
Header Name	Name of the target attachment header to compare
Comparator	The method of comparison
Header Value	The value to compare
Operation	The action to take if the task comparison is met. Actions include: <ul style="list-style-type: none"><li>• Remove: attachment will be stripped</li><li>• Block: The request will be blocked due to the existence of the attachment</li><li>• BASE64 Encode: Will encode the attachment with BASE64 encoding</li></ul>

### TASK: REMOVE TRANSPORT HEADER

The Remove Transport Header task is used to remove a header from a network transport variant (such as HTTP, JMS, etc)

## TASK LISTS > TASK LIST: NEW TASK LIST2 > TASK: REMOVE TRANSPORT HEADER

Configuration saved

### REMOVE TRANSPORT HEADER

Task Type: Remove Transport Header

Task Name\*: Remove Transport Header

Header Name\*: MyHeaderNameToRemove

## Replace Remove Transport Header Task Screen Terms

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
Header Name	The name of the protocol transport header to remove

### TASK: REMOVE WS-SECURITY HEADER

The Remove WS-Security Header task allows the system to act as a liaison between the incoming request and back end servers. With the request, the system consumes the WS-Security header, validates credentials (i.e., validates a signature), and then removes the WS-Security Header. This task is often used when the back end web server is not WS-Security-aware.

### Remove WS-Security Header Task Screen Terms

No additional settings are required, simply create and associate the task for the behavior to be active.

### Options Available When Removing a WS-Security Header

The following are options for removing the WS-Security header:

1. Using the Remove WS-Security Header task strips out the wsse:Security and wsu:Timestamp SOAP headers. This task will not remove any Id or wsu:Id attributes inserted into the document during the Sign Document task. The system functions as if in a SOAP role (or as a SOAP actor), stripping out the SOAP headers targeted at the system.
2. Using the Remove Signature checkbox in the Verify Document Signature Task will remove the WS-Security Header, any verified signature, including XML and WS-Security, and any Id or wsu:Id attributes inserted into the document during the Sign Document task. If the resulting wsse:Security header is empty, this task will strip out the wsse:Security. This task will not remove any security tokens in the wsse:Security SOAP header and will not remove the wsse:Security header if it is not empty.

## TASK: REMOVE XML NODE

The Remove XML Node task is used to remove elements from XML documents.

TASK LISTS > TASK LIST: NEW TASK LIST > TASK: REMOVE XML NODE

---

REMOVE XML NODE

---

Task Type:

Remove XML Node

Task Name\*:

On Error:

☒ Log & Halt Processing ☐ Log & Continue

---

SELECT NODES TO REMOVE

---

☐ soap:Envelope

☐ soap:Body

---

Nodes to Remove

☐ **NODE**

No items to display

Apply

### Remove XML Node Task Screen Terms

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
Select Nodes to Remove	XPath expressions that point to the nodes to remove from the document

## TASK: REPLACE DOCUMENT

The Replace Document task will replace the inbound document with the specified document. If this task is associated with a response event, then the response received from the back-end system will be replaced with the specified document.

TASK LISTS > TASK LIST: NEW TASK LIST > TASK: REPLACE DOCUMENT

---

REPLACE DOCUMENT

---

Task Type:

Replace Document

Task Name\*:

Document:

---

### Replace Document Task Screen Terms

TERM	DESCRIPTION OF OPTIONS
------	------------------------



Task Name	The name of the task
Document	The document reference from the Document policies that will be used to replace the current document from the transaction (request or response)

## TASK: TRANSFORM DOCUMENT (XSLT)

The Transform Document task uses simple or compound XSLT definitions to transform the target request or response document. XSLT style sheets used in this task may be loaded from a File or a URL and may be single definitions files, or complex XSLT with import dependencies.

**TRANSFORMATION**

Task Type:

Transform Document

Task Name\*:

Transform Document

On Error:

☒ Log & Halt Processing
☐ Log & Continue

Filename:

XSLT document\*:

☒ File
☐ URL

Choose File

No file chosen

Apply

Save

## Tranform Document Task Screen Terms

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
XSLT Document	The simple or combound set of XSLT documents that are to be used to transform the target document.

## ATTRIBUTE MAPPING TASKS

The tasks under this category are used to extract metadata properties from the request, response, or from external environment systems such as a Database or LDAP.

### TASK: MAP ATTRIBUTES AND HEADERS

The Map Attributes and Headers task is a versatile way to map information from one source to another as transactions flow through the policy. There are a variety of sources to map information from and map information to allowing complex business use case scenarios to be accomplished with simple mapping policies.

TASK LISTS > TASK LIST: MAPPINGTASKS > TASK: MAP ATTRIBUTES AND HEADERS

---

**MAPPING**

Source Type:	Request Header ▾
Source Name*:	<input type="text"/>
Target Type:	User Attribute ▾
Target Name*:	<input type="text"/>

Create

### Map Attributes and Headers Screen Terms

The sources of attributes includes

- **Constant**
  - A static value specified directly on the policy
- **Protocol Header**
  - If the task is operating on the request, this represents the request header. If the task is operating on the response, this setting represents the response header
- **Request Header**
  - The header from the inbound request from the client
- **Response Header**
  - The header from the response from the back-end system
- **User Attribute**
  - An attribute from LDAP, Active Directory, STS Identity Broker, Siteminder, or any other supported Sentry identity adapter that returns user attributes with the authentication response. Note that Identity Attributes are also used as User Attributes
- **X509 Attribute**
  - OID and other attributes within an X509 certificate
- **Query Parameter**
  - Each query string parameter from the inbound URI is available as an attribute to map
- **Cookie**
  - A cookie from the header
- **Template**
  - A variable that can be referenced within custom text or templates
- **DateTime**
  - Gets the current date and time formatted according to the input. For example, using 'Date:' yyyy-MM-dd 'Time:' HH:mm:ss.SSSz would result in the output such as Date: 2025-01-14 Time: 18:23:19.521Z
- **HTTP Method**
  - GET or POST
- **Request Path**
  - The request path of the inbound client request, not including query parameters

- **Request Path Segments**
  - Each segment of the request path able to be mapped to an
- **Request URL**
  - The request URL of the inbound client request, including path but not query parameters
- **Full Request URL**
  - The request URL of the inbound client request, including path and query parameters
- **Username**
  - The currently authenticated client's username
- **Source IP Address**
  - The current client's source IP
- **HTTP Status Code**
  - The response code from the back-end system response back to Sentry
- **HTTP Status Message**
  - The response code message from the back-end system response back to Sentry
- **Random Number**
  - A unique id generated to uniquely identify clients
- **New Session ID**
  - An id generated to uniquely identify sessions using basic HTTP authentication
- **PEM encoded X509**
  - PEM encoded X509 certificate
- **Full Document**
  - The request or response document
- **Zip Entry Name**
  - When using the Zip Contents Processing task, this source value represents the current ZIP file entry name being processing from the target ZIP file. Note that this option only appears when you have the ZIP feature enabled in your license.
- **Transaction ID**
  - Session ID that is automatically generated for the transaction

The target mapping options include:

- **Protocol Header**
  - If the task is operating on the request, this represents the request header. If the task is operating on the response, this setting represents the response header
- **Request Header**
  - The header from the inbound request from the client
- **Response Header**
  - The header from the response from the back-end system
- **User Attribute**
  - A general attribute type that can be referenced by other tasks
- **Identity Attribute**
  - A session attribute that can be used as a User Attribute and referenced by other tasks
- **Password**
  - A special attribute that will have the value obfuscated from GUI view and from the logs
- **Aggregation Attribute**
  - Allows for mapping multiple values into a single aggregation attribute. Values are comma separated
- **Query Parameter**
  - A target name value pair to add to the URI for the back-end server request from Sentry
- **Template**
  - A variable that can be referenced within custom text or templates
- **Cookie**
  - Using this setting will result in a SET-COOKIE header being created with the cookie value as the value specified.

- **HTTP Method**
  - GET or POST
- **Request Path**
  - The request path of the inbound client request, not including query parameters
- **HTTP Status Code**
  - The response code from the back-end system response back to Sentry
- **Full Document**
  - Overwrites the current request or response body with the source value
- **Zip Entry Name**
  - When using the Zip Contents Processing task, this value represents the current ZIP file entry name being processing from the target ZIP file as a target to replace the name. Note that this option only appears when you have the ZIP feature enabled in your license.

## TASK: MAP ATTRIBUTES FROM XML

The Map Attributes from XML task allows you to extract information from an XML/SOAP document and map these values into attributes which can be used and referenced by other tasks or map to other policy locations.

The screenshot shows the 'TASK: MAP ATTRIBUTES FROM XML' configuration screen in the ForumSentry API Security Gateway. The interface includes a left sidebar with navigation menus (GENERAL, DIAGNOSTICS, GATEWAY) and a main content area. The main area displays the task configuration for 'MAP ATTRIBUTES FROM XML', including fields for Task Type, Task Name, Map To (a dropdown menu), On Error (a radio button), and a list of elements to map. The dropdown menu is open, showing options like Request Header, Response Header, User Attribute, Identity Attribute, Aggregation Attribute, Query Parameter, Cookie, Template, HTTP Method, Request Path, HTTP Status Code, and Zip Entry Name. The 'REQUIRED USER ATTRIBUTE' section is also visible.

## Map Attributes from XML Task Screen Terms

Mapping options include:

- **Protocol Header**
  - If the task is operating on the request, this represents the request header. If the task is operating on the response, this setting represents the response header
- **Request Header**
  - The header from the inbound request from the client
- **Response Header**
  - The header from the response from the back-end system
- **User Attribute**
  - A general attribute type that can be referenced by other tasks
- **Identity Attribute**

- A session attribute that can be used as a User Attribute and referenced by other tasks
- **Aggregation Attribute**
  - Allows for mapping multiple values into a single aggregation attribute. Values are comma separated
- **Query Parameter**
  - A target name value pair to add to the URI for the back-end server request from Sentry
- **Template**
  - A variable that can be referenced within custom text or templates
- **Cookie**
  - Using this setting will result in a SET-COOKIE header being created with the cookie value as the value specified.
- **HTTP Method**
  - GET or POST
- **Request Path**
  - The request path of the inbound client request, not including query parameters
- **HTTP Status Code**
  - The response code from the back-end system response back to Sentry

## TASK: MAP ATTRIBUTES TO XML

The Map Attributes to XML task allows you to set or insert attributes coming from a range of difference sources, and map these attributes into XML document elements.

The screenshot displays the 'TASK: MAP ATTRIBUTES TO XML' configuration page within the Forum Sentry API Security Gateway. The interface is divided into a left sidebar and a main content area.

**Left Sidebar:**

- GENERAL**
  - Forum Systems
  - Getting Started
  - Help
- DIAGNOSTICS**
- GATEWAY**
  - Network Policies
  - Network Policies
  - Proxy Policies
  - Cloud Policies
  - Cache Policies
- WSDL Policies**
  - WSDL Libraries
  - WSDL Policies
- Content Policies**
  - XML Policies
  - REST Policies
  - JSON Policies
  - HTML Policies
  - STS Policies
  - OAuth Policies
  - Tests
- Task Policies**
  - Task List Groups
  - Task Lists
- Redirect Policies**
  - Redirect Policies
- Request Filters**
  - Request Filters
- RESOURCES**
- IDP**
- ACCESS**
- SYSTEM**
- PARTNERS**

**Main Content Area:**

**TASK LISTS > TASK LIST: ALERT TASK > TASK: MAP ATTRIBUTES TO XML**

**MAP ATTRIBUTES TO XML**

Task Type: Map Attributes to XML

Task Name\*: Map Attributes to XML

Map From: User Attribute

On Error: Constant

**SELECT TARGET ELEMENTS**

☐ soap:Envelope

☐ soap:Body

**Target Document Elements**

☐ ELEMENT

No items to display

**USER ATTRIBUTE**

Protocol Header

Request Header

Response Header

User Attribute

X.509 Attribute

Query Parameter

Cookie

Template

DateTime

HTTP Method

Request Path

Request URL

Username

Source IP Address

HTTP Status Code

Random Number

New Session Id

PEM encoded X509

Zip Entry Name

## Map Attributes to XML Task Screen Terms

The sources of attributes includes

- **Constant**
  - A static value specified directly on the policy
- **Protocol Header**
  - If the task is operating on the request, this represents the request header. If the task is operating on the response, this setting represents the response header
- **Request Header**
  - The header from the inbound request from the client
- **Response Header**
  - The header from the response from the back-end system
- **User Attribute**
  - An attribute from LDAP, Active Directory, STS Identity Broker, Siteminder, or any other supported Sentry identity adapter that returns user attributes with the authentication response. Note that Identity Attributes are also used as User Attributes
- **X509 Attribute**
  - OID and other attributes within an X509 certificate
- **Query Parameter**
  - Each query string parameter from the inbound URI is available as an attribute to map
- **Cookie**
  - A cookie from the header
- **Template**
  - A variable that can be referenced within custom text or templates
- **DateTime**
  - Gets the current date and time formatted according to the input. For example, using 'Date:' yyyy-MM-dd 'Time:' HH:mm:ss.SSSz would result in the output such as Date: 2025-01-14 Time: 18:23:19.521Z
- **HTTP Method**
  - GET or POST
- **Request Path**
  - The request path of the inbound client request, not including query parameters
- **Request Path Segments**
  - Each segment of the request path able to be mapped to an
- **Request URL**
  - The request URL of the inbound client request, including path but not query parameters
- **Full Request URL**
  - The request URL of the inbound client request, including path and query parameters
- **Username**
  - The currently authenticated client's username
- **Source IP Address**
  - The current client's source IP
- **HTTP Status Code**
  - The response code from the back-end system response back to Sentry
- **HTTP Status Message**
  - The response code message from the back-end system response back to Sentry
- **Random Number**
  - A unique id generated to uniquely identify clients
- **New Session ID**
  - An id generated to uniquely identify sessions using basic HTTP authentication
- **PEM encoded X509**
  - PEM encoded X509 certificate
- **Full Document**
  - The request or response document

- **Zip Entry Name**
  - When using the Zip Contents Processing task, this source value represents the current ZIP file entry name being processing from the target ZIP file. Note that this option only appears when you have the ZIP feature enabled in your license.
- **Transaction ID**
  - Session ID that is automatically generated for the transaction

The Map Attribute to XML task provides the means to extract information in the form of attributes from various sources and map these values into the XML/SOAP document that is to be returned to the client, or proxied to the back-end system.

For simplicity, it is recommended that a sample document be loaded in to the Document section and then use this sample document when creating the Map Attributes to XML task. This enables graphical selection of the element nodes to map the data into.

Users have the option to read attributes from the following sources and map them to specific nodes within the XML document:

## TASK: MAPPING TABLE

The Mapping Table task allows for mapping an attribute to a lookup table in order to find a corresponding associated value from the table. The mapping table feature allows the definition of name/value pairs to define the table, and then the ability to leverage a source and target attribute to use for the lookup and setting the resulting value to an attribute.

TASK LISTS > TASK LIST: NEW TASK LIST > TASK: MAPPING TABLE

MAPPING TABLE

Task Type:

Mapping Table

Task Name\*:

Mapping Table

CONFIGURATION

Require attribute mapping to exist (fail if no key found):

☐

Source Attribute (to match to the table key)

lookupKeyAttribute

Destination Attribute (created from the value of the matched key):

destinationAttribute

LOOKUP TABLE

Lookup table:

(define table with name=value entries, 1 per line)

1=a  
2=b  
3=c  
4=d

Apply

Save

## Mapping Table Task Screen Terms

While using the Mapping Table task, please consider the following terms and definitions:

TERM	DESCRIPTION OF OPTIONS
------	------------------------

Task Name	The name of the task
Require attribute mapping to exist (fail if no key found)	When checked, requires that the mapping must succeed, or the task processing will return a failure
Source Attribute (to match the table key)	This is the name of the attribute holding the value to match against the table (i.e. the attribute value used to lookup the "name" in the name=value table entry)
Destination Attribute (created from the value of the matched key)	The target attribute to set with the value found in the lookup table. The attribute value will be set to the "value" defined in the name=value table entry if a match was found.
Lookup Table	The lookup table that is used. The key and values are defined as name=value, 1 per line

## TASK: QUERY DATABASE

The Query Database task is used to run queries against target data sources (defined under Logging->Data Sources) and use the results for mapping to other locations or to build XML documents automatically.

TASK LISTS > TASK LIST: MAPPING ATTRIBUTES > TASK: QUERY DATABASE

QUERY DATABASE

Task Type: Query Database

Task Name\*: Query Database

On Error: ☒ Log & Halt Processing ☐ Log & Continue

SQL:

Data Source: Database\_Policy [Edit](#)

Output: XML

Result Set Attribute Key Prefixes: XML

Legacy Output Mode: Attributes

SQL Values (Click To Remove)

<input type="checkbox"/> SQL PARAMETER MODE	SQL TYPE	SOURCE TYPE	SOURCE NAME	OUTPUT NAME
<a href="#">Apply</a> <a href="#">Save</a>				

### Query Data Source Task Screen Terms

While using the Query Data Source task, please consider the following terms and definitions:

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task



SQL	The SQL query to run against the data source policy. SQL can contain '?' characters for dynamic substitution with variables specified under SQL Values. To see the SQL values appear, press "Apply" button when the SQL contains '?' characters.
Data Source	The target Data Source policy that contains the information about the database
Output	<ul style="list-style-type: none"> <li>• <b>XML</b> <ul style="list-style-type: none"> <li>○ Automatically creates an XML document from the query response</li> </ul> </li> <li>• <b>Attributes</b> <ul style="list-style-type: none"> <li>○ Creates User Attribute type mappings under the names Table.Field for each column response. These values can then be used in Mapping tasks to map this information elsewhere.</li> </ul> </li> </ul>

## Behavior

- **Dynamic Placeholders:** When the Aggregate Attribute is chosen, additional placeholders are automatically inserted into the SQL statement. These placeholders correspond to the individual values within the aggregate attribute.
- **Prepared Statement Binding:** The corresponding values from the aggregate attribute are individually set within the prepared statement, ensuring security and efficiency.
- **IN Clause Handling:** For IN clauses, the functionality seamlessly adds separate placeholders for each value in the list.
- **OR Condition Detection:** If the functionality detects a condition within the user-provided values, it constructs an OR clause to incorporate those conditions into the SQL statement.

### Example 1: IN Clause Expansion

#### User Input:

- SQL Statement:

SELECT \* FROM Users WHERE firstName in (?)

- Values: 'John,Alice,Sophia' (comma-separated string)
- **Generated SQL:**

SELECT \* FROM Users WHERE firstName in (?, ?, ?)

- The values ('John', 'Alice', 'Sophia') are bound to the three individual placeholders in the prepared statement.

### Example 2: OR Condition Handling

#### User Input:

- **SQL Statement:**

```
SELECT * FROM cities WHERE City=?
```

- **Values:** 'Rome','Berlin' (comma-separated string)
- **Generated SQL:**

```
SELECT * FROM cities WHERE City='Rome' OR City='Berlin'
```

The functionality detects that the user provided multiple values for the City parameter and automatically creates an OR condition to handle them.

*Usage* Select the Aggregate Attribute as the source type for the parameter. Provide your SQL statement with a single placeholder for the aggregate attribute. Configure your list of values in an aggregate attribute or other suitable format like a list. The DB Query Task will automatically generate the appropriate SQL statement with placeholders and handle the prepared statement binding for each value.

## TASK: QUERY LDAP

The Query LDAP task allows for LDAPv3 attributes an LDAP repository to be manipulated via Forum Sentry. The task allows for LDAP attributes to be read, added, replaced or removed.

TASK LISTS > TASK LIST: LDAP-MULTIPLEATTRIBUTES > TASK: QUERY LDAP

---

**QUERY LDAP**

Task Type:	Query LDAP
Task Name*:	<input type="text" value="Query LDAP"/>
On Error:	<input checked="" type="radio"/> Log & Halt Processing <input type="radio"/> Log & Continue
Output:	<input checked="" type="radio"/> Read Attributes <input type="radio"/> Add Attributes <input type="radio"/> Replace Attributes <input type="radio"/> Remove Attributes
LDAP Policy*:	<input type="text" value="QA-LDAP"/> <a href="#">Edit</a>
Search Attribute*:	<input type="text" value="lookupValue"/>
Search Attribute Type:	<input type="radio"/> Username <input checked="" type="radio"/> Email <input type="radio"/> Distinguished Name
Attribute Names*:	<input type="text" value="cn,sn,uid,mail,employeeType,employeeNumber"/>

### Query LDAP Task Screen Terms

While using the Query LDAP Source task, please consider the following terms and definitions:

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
Output	Specifies whether the action is to read, add, replace or remove the user identify attributes.
LDAP Policy	The target LDAP policy that contains the information about how to connect and authenticate to the LDAP instance
Search Attribute	The user attribute defined that contains the data to be searched for in LDAP. <b>The user attribute should be created and defined before using the Query LDAP task.</b>  For example, if you wanted to search LDAP for an email address. A Sentry user attribute called <b>lookupValue</b> is created with value <a href="#">user@company.com</a> in an earlier task.  Then this task would use “Read Attributes” for the Output setting and <b>lookupValue</b> as the Search Attribute and the Search Attribute Type would be Email)
Search Attribute Type	The type of attribute being search for in LDAP. This can either be a username, email address or distinguished name (e.g. e.g. <a href="#">search-dn=user@company.com</a> where the type in this case is Email)
Attribute Names	Specifies the name of the attributes for the Query LDAP task action.

## USER IDENTITY AND ACCESS CONTROL TASKS

The tasks under this category are used to restrict access based on IP address or username and control how the users are authenticated and authorized.

### TASK: IP ACL

The Task IP ACL is used to apply IP based access control when processing the task list. The IP of the source request is used to evaluate against the selected IP ACL policy to apply the allow or deny rule.

If the associated IP ACL policy triggers a deny event and the "Store as attribute" is not set, then the task will fail and trigger an IDP error message. If the "Store as attribute" is enabled, then the success or failure status of applying the IP ACL will be stored in the user attribute. The user attribute stores the values "success" or "fail" depending on if the application of the ACL succeeded or failed.

TASK LISTS > TASK LIST: NEW TASK LIST2 > TASK: IP ACL

---

IP ACL

---

Task Type:	IP ACL
Task Name*:	<input type="text" value="IP ACL"/>
IP ACL Policy:	<div>Unrestricted <input type="button" value="Edit"/></div>
Store User Attribute:	<input type="checkbox"/>
User Attribute:	<input type="text"/>

---

### IP ACL Task Screen Terms

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
IP ACL Policy	The IP ACL Policy from Access->IP ACLs to apply
Store User Attribute	When enabled, if the associated IP ACL policy triggers a deny event the success or failure status of applying the IP ACL will be stored in the user attribute
User Attribute	The user attribute stores the values "success" or "fail" depending on if the application of the ACL succeeded or failed.

### TASK: LOGOUT

The Logout task is used for invalidating the session token for SiteMinder or the session Token for Forum Sentry or Forum STS tokens. This task requires persistent session caching is enabled on the SiteMinder policy server, or persistent sessions enabled on Forum Sentry or Forum STS Identity Broker. Upon receipt of a message containing a session token, the system will send the token to the aforementioned identity server to invalidate the session.

## Logout Task Screen Terms

No additional settings are required, simply create and associate the task for the logout behavior to be active.

## TASK: USER IDENTITY AND ACCESS CONTROL

The User Identity and Access Control Task allows Administrators to designate an Access Control List (ACL) for a given Task List, and establish an identity for the user. The identity is derived from the protocol using a standard, such as HTTP Basic Authentication or from the message itself via a SAML Assertion or a WS-Security header.

**TASK LISTS > TASK LIST: MAPPING ATTRIBUTES > TASK: USER IDENTITY & ACCESS CONTROL**

---

**USER IDENTITY MECHANISM**

- ☐ Identity established in network policy (basic auth or client cert)
- ☐ Identity established by validating cookies
- ☐ Validate WS-Security & establish identity
- ☐ Validate SAML assertion & establish identity
- ☐ Validate SAML SSO assertion & establish identity
- ☒ Validate OAuth token & establish identity
- ☐ Validate OAuth SSO token & establish identity
- ☐ Identity established by attribute mapping
- ☐ Identity established by digital signature
- ☐ Identity established by Sentry REST authentication

---

**Next**

---

**USER IDENTITY & ACCESS CONTROL**

Task Type: User Identity & Access Control

Task Name: User Identity & Access Control

ACL Policy: Allow All

Configuration options available with User Identity and Access Control include:

The specifications supported on the system for the User Identity & Access Control task are:

- HTTP 1.0/1.1
- SSLv3, TLS 1.0
- WS-Security 1.1
- WS-Security 2004
- WS-Security 2004 Kerberos Token Profile 1.1
- WS-Security 2004 SAML Token Profile 1.1

- WS-Security 2004 Username Token Profile 1.1
- WS-Security 2004 X.509 Certificate Token Profile 1.1
- SAML 1.0, 1.1, and 2.0
- SAML 2.0 WEB SSO Profile
- OAuth 2.0

### Access Control Lists

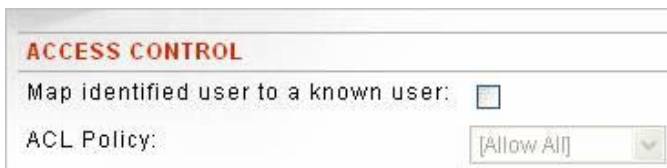
Access Control Lists (ACL) are sets of user groups which have either been granted or denied access to a Task List. The User Identity & Access Control task also allows Administrators to designate an Access Control List (ACL) for a given lists of Tasks, and establish an identity for the user. The identity is derived from the protocol using a standard, such as HTTP Basic Authentication, SSL client authentication or URI authentication. SSL Protocol Authentication provides X.509 path processing validation.

Credential binding is an authentication mechanism that is used for document processing. The credentials that can be bound are Username/Password, X.509 DN and SAML Assertion.

### Access Control Options with User Identity and Access Control Tasks

The options available when applying access control to a User Identity & Access Control task are:

- **No access control is active** during this User Identity and Access Control task. The user is identified from the protocol or the document but is not matched to any known user in Forum or in any third party user store. For example, if the user provided an X.509 certificate, Forum may verify that the certificate is valid and identify the subject of the certificate as the user, but Forum does not in any way restrict the set of allowed users.

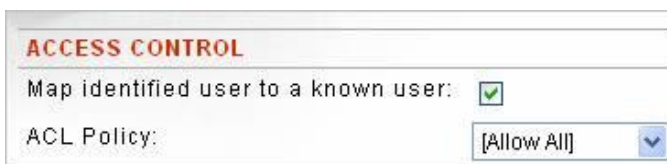


**ACCESS CONTROL**

Map identified user to a known user: ☐

ACL Policy: [Allow All]

- **No Forum ACL is active** during this User Identity and Access Control task. The user is identified from the protocol or the document and is matched to a known user in Forum or in a third party user store. The user is not restricted by any Forum ACL.

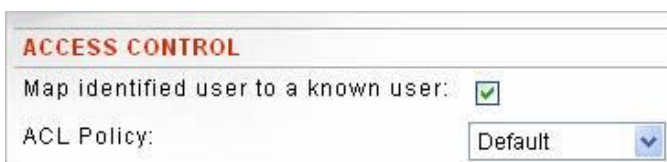


**ACCESS CONTROL**

Map identified user to a known user: ☒

ACL Policy: [Allow All]

- **Forum ACL is active** during this User Identity and Access Control task. The user is identified, matched to a known user, and restricted by Forum ACL.



**ACCESS CONTROL**

Map identified user to a known user: ☒

ACL Policy: Default

## Prerequisites for All User Identity and Access Control Tasks

Before performing any of these operations listed above, except for No access control, it is assumed that:

- at minimum, one User, Group and ACL have been created in the Users, Groups and ACLs sections of the WebAdmin.
- this user has been assigned membership into a Group (from the User Details screen or from the Groups screen), and the Group has been assigned membership into the ACL from the ACL Policy screen of the WebAdmin. For more information, refer to the Users, Groups and ACLs sections of the *Forum Systems Sentry™ Version 9 Access Control Guide*.

## User Identity and Access Control Task Screen Terms

The following table displays all the terms and definitions found in the User Identity and Access Control task wizard:

TERM	DEFINITION
Task Name	The name given to this task. Users may accept the default task name or give the task a unique name.
Map Identified user to a known user	<ul style="list-style-type: none"><li>• When checked, the user credentials are mapped to a known user configured in the system or in an external user store.</li><li>• When unchecked, the user credentials are not mapped to a known user.</li></ul>
Access Control	<p>The name of the access control list to apply to this task.</p> <p>For more information, refer to the Access Control Options with User Identity and Access Control tasks section discussed earlier in this chapter.</p>
User Identity Mechanism	<ul style="list-style-type: none"><li>• With Identity established in network policy (password auth or client cert) selected, the user in the document is identified by protocol authentication such as HTTP Basic Auth or SSL.</li><li>• With Validate WS-Security and establish identity selected, the user is identified from a security token in the WS-Security header.</li><li>• With Validate SAML assertion and establish identity selected, the user in the document is identified from a SAML assertion.</li><li>• With Validate SAML SSO assertion and establish identity selected, the SAML Web SSO profile is configurable as to whether to use SP-Initiated, or iDP initiated SAML Web SSO profile to authenticate the user via HTTP redirects</li><li>• With Identity established by OAuth, the OAuth credentials are extracted and validated</li><li>• With Identity established by XML mapping selected, the username and password values entered in the Mapped Attributes dialog are used to identify the user in the document.</li><li>• With Identity established by digital signature selected, the user is identified based on the X.509 certificate used by a required prior Verify Document Signature task to verify a digital signature in the document.</li></ul>
Security Token Type	<ul style="list-style-type: none"><li>• With Username token selected, a Username token is required in the document for user identification.</li><li>• With X.509 binary token selected, an X.509 binary token is required in the document for user identification.</li><li>• With SAML token selected, a SAML assertion required in the document for user identification.</li><li>• With Kerberos token selected, a Kerberos token is required in the document for</li></ul>

user identification.	
Require Password	With identity established by Username token, requires the password of the user being identified.
Issuer(s)	The Validate issuer by name checkbox (optional) specifies which issuer(s) are allowed. If specified, the issuer name should match the issuer name used by the sender (e.g. as configured in the SAML Assertion or WS-Security Header task).
TERM	DEFINITION
Verification Policy	The Verification policy to use when verifying the signature when establishing the identity by SAML.
Require Signature	The Require signature checkbox verifies that the assertion is signed and that the signature is valid.
SAML Identity Mechanism	<ul style="list-style-type: none"> <li>With Email selected, user identity is established by the Email address inside a SAML assertion.</li> <li>With X.509 DN selected, user identity is established by the X.509 DN inside a SAML assertion.</li> <li>With Attribute selected, user identity is established by the value of the attribute specified in the Attribute dialog.</li> </ul>
Mapped Attributes	<p>Username attribute is the attribute to which the username was mapped from the document in the proceeding Map Attributes from XML task.</p> <p>Password attribute is the attribute to which the optional password was mapped from the document in the proceeding Map Attributes from XML task.</p>

### Protocol-based User Identity and Access Control

Access control and authorization is supported on the system for transport-centric mechanisms such as HTTP Basic Auth and SSL Client Certificates. Follow these steps to authenticate a user and allow access by adding the Identity and Access Control task by HTTP Protocol.

TASK NAME

Task Name\*:

Next

ACCESS CONTROL

Map identified user to a known user: ☒

ACL Policy:

Next



## USER IDENTITY MECHANISM

- ☒ Identity established in network policy (basic auth or client cert)
- ☐ Identity established by validating cookies
- ☐ Validate WS-Security & establish identity
- ☐ Validate SAML assertion & establish identity
- ☐ Validate SAML SSO assertion & establish identity
- ☐ Validate OAuth token & establish identity
- ☐ Validate OAuth SSO token & establish identity
- ☐ Identity established by attribute mapping
- ☐ Identity established by digital signature
- ☐ Identity established by Sentry REST authentication

**Next**

- From the TASK screen, select **New**.
- On the TASK TYPE screen, select the **User Identity & Access Control** radio button, and then click **Next**.
- On the TASK NAME screen, accept the default task name or enter a **task name**, and then click **Next**.
- On the ACCESS CONTROL screen, check the **Map identified user to a known user** checkbox.
- Select an **ACL** from the ACL Policy drop down list, and then click **Next**.
- On the USER IDENTITY MECHANISM screen, select the **Identity established in server policy** checkbox, and then click **Next**.
- On the Error Template screen click **Finish**.

### Add User Identity and Access Control by XML Mapping Task

During the User Identity/Access Control by XML Mapping task, the user is identified based on the username and password in the document.

The actual attribute names used in this task can be anything as long as the same attribute names are specified in both tasks and the specified xml elements contain the actual username and password. The password may be omitted in both tasks if no password checking is required.

### Add User Identity and Access Control by Digital Signature Task

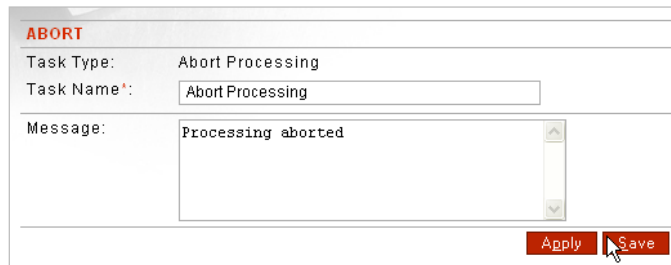
During the User Identity/Access Control by Digital Signature task, using the Establish identity by digital signature option, the user is identified based on the X.509 certificate used by a prior Verify Document Signature task to verify a digital signature in the document. This task assumes an XML Verification Policy exists for the user.

## FLOW CONTROL TASKS

The tasks under this category are used to control the aspects of the transaction ranging from abort to redirection, dynamic routing, or iteration tasks.

## TASK: ABORT PROCESSING

When selected, the Abort Processing task halts processing of the document and returns a specified message to the client. No additional tasks in the Task List will be processed.

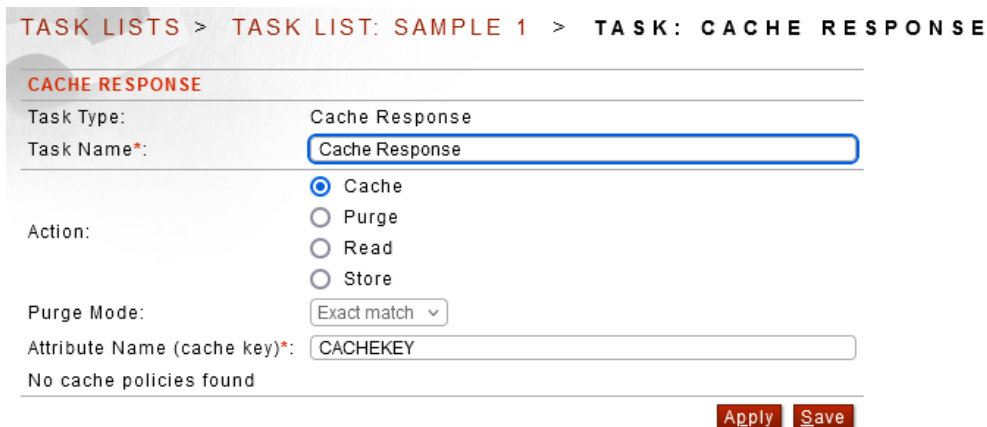


### Abort Task Screen Terms

No additional settings are required, simply create and associate the task for the behavior to be active.

## TASK: CACHE RESPONSE

This task enables granular ability to utility a caching policy



### Cache Response Task Screen Terms

TERM	DEFINITION
Task Name	The name given to this task. Users may accept the default task name or give the task a unique name.
Action	<ul style="list-style-type: none"><li>• Cache</li><li>• Purge</li><li>• Read</li><li>• Store</li></ul>
Purge Mode	<ul style="list-style-type: none"><li>• Exact Match</li><li>• Starts With</li></ul>
Cache Key	Attribute Name which has the cache key to use
Cache Policy	The cache policy to use for this task

## TASK: DELAY PROCESSING

This task enables the Sentry policy to induce the specified amount of latency to the transaction. This can be used in cases where for testing purposes different latency characteristics need to be measured, or in cases where the clients are meant to be queued at a distinct rate for getting information from the back-end system.

TASK LISTS > TASK LIST: JSON TASKS > TASK: DELAY PROCESSING

DELAY

Task Type:	Delay Processing
Task Name*:	<input type="text" value="Delay Processing"/>
Delay(ms):	<input type="text" value="0"/>

Apply

### Delay Processing Screen Terms

While using this task , please consider the following terms and definitions:

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
Delay (ms)	The amount of additional latency (wait time) to induce to the policy task list processing.

## TASK: FOR LOOP

This task enables iterating through a series of values associated with an aggregate user attribute. For each iteration, the associated Task List or Task List Group will be invoked with the current state of the request or response transaction context where the task is invoked. The result user attribute can store the final iteration result. This task can be run synchronously (recommended) or asynchronously.

Note: Be very cautious using this task in asynchronoous mode as it can significantly impact the Sentry system resource consumption and thread allocation.

**FOR LOOP**

Task Type:	For Loop		
Task Name*:	<input type="text" value="For Loop"/>		
On Error:	<input checked="" type="radio"/> Log & Halt Processing <input type="radio"/> Log & Continue		
Item User Attribute*:	<input type="text" value="item"/>		
Items User Attribute*:	<input type="text"/>		
Processing*:	<input type="text" value="Task Lists"/>	<input type="text" value="[None]"/>	
Asynchronous:	<input type="checkbox"/>		
Result User Attribute*:	<input type="text"/>		

**For Loop Screen Terms**

While using this task , please consider the following terms and definitions:

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the Task
Item User Attribute	The attribute that will store each iterator value extracted from Items User Attribute
Items User Attribute	The Aggregate User Attribute that contains the set of iterator values
Processing	The Task List or Task List Group to run for each extracted iterator value
Aynchronous	Whether or not to run this loop synchronously (each iteration follows the next one concurrently), or asynchronously (each iteration runs immediately and independent of the others)
Result User Attribute	The attribute to store the end result of the loop iteration processing

The For Loop task allows you to iterate over a list of values (provided via a user attribute), executing a specified sub-process (like a Task List) for each value in the list. This task is useful for scenarios requiring repetitive processing over a collection of data, such as processing multiple users, files, or request elements.

**Configuration Parameters**

- **Items Attribute (itemsAttribute):** The name of the user attribute containing the list of items to iterate over. This can be a String[], a comma-separated String, or a JSONArray.
- **Item Attribute (itemAttribute):** The name of the user attribute where the current item from the list will be placed for each iteration. Defaults to "item".
- **Process (process):** The name of the sub-process (e.g., Task List name) to execute for each item.
- **Process Type (processType):** The type of the sub-process to execute. Defaults to TASK\_LIST.
- **Asynchronous (asynchronous):**
  - false (Synchronous Mode - Default): Each iteration completes fully before the next one begins.

- true (Asynchronous Mode): Iterations are launched concurrently (potentially limited by system resources). The main loop proceeds without waiting for each sub-process to finish individually.
- **Result Attribute (resultAttribute):** (Optional, primarily for Asynchronous mode) If specified, the task collects the value of this attribute from each completed iteration's ProcessState and stores the results as a String[] in the original ProcessState under this attribute name *after* all launched asynchronous iterations have been waited for.
- **Break Attribute Name (breakAttributeName):** (Optional) The name of a user attribute whose value will be checked *before* each iteration to potentially stop the loop early.
- **Break Attribute Value (breakAttributeValue):** (Optional) The specific string value that the attribute specified by breakAttributeName must have to trigger the loop break. **Defaults to "true".**

## Iteration Logic

1. The task retrieves the value of the Items Attribute.
2. It determines the list of items based on the type (String[], CSV String, JSONArray).
3. If the Items Attribute value is **not** an array/list/CSV (or is null), the sub-process is executed **only once**, using the original value as the Item Attribute. **The break condition is NOT checked in this single-execution case.**
4. If it's a list of items, the task iterates through them.
5. **Before** executing the sub-process for an item, the task checks the break condition (see below).
6. If the break condition is met, the loop terminates immediately, and no further items are processed.
7. If the break condition is not met, the current item is placed in the Item Attribute within the appropriate ProcessState (original state for sync, a clone for async).
8. The specified Process is executed using that ProcessState.
9. The loop continues to the next item (or finishes if it was the last item).

## Break Condition Functionality

- The break condition is only evaluated if Break Attribute Name is configured (not null or empty).
- The check occurs **before** each potential iteration begins (including the very first one).
- The task reads the value of the attribute named by Break Attribute Name from the ProcessState relevant to the *upcoming* check.
  - In **Synchronous mode**, this is always the main originalProcessState, which may have been modified by the *previous* iteration.
  - In **Asynchronous mode**, this is the ProcessState object that was used (cloned) by the *previous* iteration.
- The loop breaks if **all** of the following are true:
  1. The attribute specified by Break Attribute Name exists in the checked ProcessState.
  2. The attribute's value is **not** null.
  3. The **string representation** of the attribute's value exactly matches the configured Break Attribute Value (which defaults to "true").
- If the loop breaks, a FINE level log message (TASK\_FOR\_LOOP\_BREAKING) is generated.

## Async vs. Sync Modes (Break Behavior)

The core break logic (checking the condition based on the relevant state before the next iteration) works similarly in both modes, leading to immediate termination of *further* loop progression:

- **Synchronous Mode:** The loop executes sequentially. When the break condition is met (based on the state *after* the previous iteration completed), the loop stops *before* the next iteration's sub-process is called.

- **Asynchronous Mode:** Iterations are launched concurrently. When the break condition is met (based on the state of the *clone* used by the previous iteration), the loop stops *before* launching the *next* asynchronous sub-process task.
  - **Important:** Any asynchronous tasks already launched *before* the break condition was met will continue to run and complete independently.
  - If Result Attribute is configured, the task will still wait for the already-launched tasks to complete, but the final result list will only contain results from iterations that were launched *before* the break occurred.

### Example Scenario (Break Condition)

#### 1. ForLoop Configuration:

- itemsAttribute = "userList" (contains ["A", "B", "C", "D"])
- itemAttribute = "currentUser"
- process = "ProcessSingleUser"
- breakAttributeName = "stopProcessing"
- breakAttributeValue = "1" (Note: default is "true")
- asynchronous = false (Sync Mode)

#### 2. Execution Flow:

- **Before Item "A":** Check ProcessState for stopProcessing. Assume not found -> No Break. Run "ProcessSingleUser" with currentUser="A". Assume this run does *not* set stopProcessing.
- **Before Item "B":** Check ProcessState for stopProcessing. Not found -> No Break. Run "ProcessSingleUser" with currentUser="B". Assume this run *sets* stopProcessing="1" in the ProcessState.
- **Before Item "C":** Check ProcessState for stopProcessing. Found, value is "1". This matches breakAttributeValue. **Break the loop.**
- Items "C" and "D" are **not** processed.

## TASK: REDIRECT

This task provides the ability to issue a redirect back to the calling client with a redirect code which the client application or browser will handle as expected by HTTP specification.

TASK LISTS > TASK LIST: SAMPLE 2 > TASK: REDIRECT

---

**REDIRECT TASK**

Task Type: Redirect

Task Name\*:

Redirect Code:

Source Type:

Source Name\*:

### Redirect Example

You can see an example of this task to convert from HTTP to HTTPS on our Helpdesk at [Redirect HTTP Requests to use HTTPS – Forum Systems Support](#).

### Redirect Screen Terms

While using this task , please consider the following terms and definitions:

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the Task
Redirect Code	Codes 301-307
Source Type	The Aggregate User Attribute that contains the set of iterator values
Processing	The Task List or Task List Group to run for each extracted iterator value
Aynchronous	Whether or not to run this loop synchronously (each iteration follows the next one concurrently), or asynchronously (each iteration runs immediately and independent of the others)
Result User Attribute	The attribute to store the end result of the loop iteration processing

## TASK: REMOTE ROUTING

This task provides options for routing the message based on content or to make asynchronous or synchronous copies of the inbound document to send the copy to a target service for processing while still processing the original request.

### Content-based Routing Using the Remote Routing Task

Content-based routing provides a method of overriding a remote policy, a remote path or both for a request or response. With both WSDL and XML policies, Administrators may configure an HTTP/S, Tibco-Rv, Tibco-EMS, or MQ policy with the Remote Routing task to re-route the document to a specified remote policy; and in the case of HTTP/S policies, to a specified remote path. Additionally, users may set a specific action to apply to the remote routing that document will follow while being processed.

Administrators applying the Remote Routing task have the following options:

- Select the **Override remote routing** action.
  - Check the **Remote Policy** checkbox (which retains the same remote path).
  - Check the **Remote Policy** and the **Remote Path** checkboxes.
  - Check the **Remote Path** checkbox (which retains the same back end server).
- Select the **Send asynchronous message copy** action.
- Select the **Send synchronous message copy** action.
- Select the **Replace message with remote response** action.

### Remote Routing Screen Terms

The following table displays the terms and definitions found in the Remote Routing screen:

TERM	DEFINITION
Task Name	Identifier for this task.
Action	<ul style="list-style-type: none"><li>• With <b>Override remote routing</b> selected, the remote server for the request being processed is changed to the selected Remote Policy. Additionally, HTTP/S policies may override the Remote Path.</li><li>• With <b>Send asynchronous message copy</b> selected, the system sends a copy of the document as it exists at that point in processing to the remote server asynchronously, using a new doc Id. The system proceeds immediately to processing the next Task in the Task List. When the asynchronous response is received, it is logged but not used for further processing in the foreground Task List.</li><li>• With <b>Send synchronous message copy</b> selected, the system sends a copy of the document as it exists at that point in processing to the remote server synchronously. The system waits for a response from the remote server. If successful, the system continues processing the next Task in the Task List. If there is an error from the remote server, then all task processing is halted.</li><li>• With <b>Replace message with remote response</b> selected, the system sends a copy of the document as it exists at that point in processing to the remote server synchronously. The system waits for a response from the remote server. If successful, the system replaces the document that is being processed with the response, and continues processing the next Task in the Task List. If there is an error from the remote server, then processing is halted.</li></ul>
Remote Policy	With Remote Policy selected, re-route the document to a specified remote policy.
Remote Path	With Remote Path supplied, re-route the document to a specified remote path.



## TASK: WS-ADDRESSING

The Sentry WS-Addressing task supports the OASIS WS-Addressing specification for both synchronous and asynchronous messaging paradigms. This task can be used for dynamic routing as well as providing asynchronous long running transaction support.

TASK LISTS > TASK LIST: NEW TASK LIST2 > TASK: WS-ADDRESSING

### WS-ADDRESSING

Task Type:	WS-Addressing
Task Name*:	<input type="text" value="WS-Addressing"/>
On Error:	<input checked="" type="radio"/> Log & Halt Processing <input type="radio"/> Log & Continue
Mode:	<input checked="" type="radio"/> Process WS-Addressing headers <input type="radio"/> Process asynchronous response <input type="radio"/> Set WS-Addressing headers
Action:	<input type="text"/>
<input checked="" type="checkbox"/> Route to destination	
<input type="checkbox"/> Allow anonymous destination	
<input type="checkbox"/> Allow asynchronous response	
<input checked="" type="radio"/> Reply to listener policy	<input type="text" value="HttpListenerPolicy-2 (0.0.0.0:8097)"/> <a href="#">Edit</a>
<input type="radio"/> Specify reply address	
Reply Protocol:	<input type="text" value="http"/>
Reply Host:	<input type="text"/>
Reply Port:	<input type="text" value="80"/>
<input checked="" type="checkbox"/> Asynchronous timeout	<input type="text" value="120"/> seconds
<input type="checkbox"/> Persistent message tracking	<input type="text" value="MySQL_Local"/> <a href="#">Edit</a>

### ALLOWED DESTINATION URLS

\*

### ALLOWED REPLY URLS

\*

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

## WS-Addressing Task Screen Terms

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
OnError	Allows the task to proceed to the next task if there is an error, or throw control to the IDP framework Process Error otherwise
Process WS-Addressing Headers	Performs replacement on the headers as applicable per the intermediary
Process Asynchronous Response	Enables the stateful persistence of the expected ReplyTo to handle a subsequent response of this conversation
Set WS-Addressing Headers	Enables creation and configuration of additional WS-Addressing Headers
Action: Route to Destination	Use the WS-Addressing headers to dynamically determine where to send the address to
Action: Allow anonymous destination	Allow the value of the the destination to be anonymous
Action: Allow Asynchronous Response	Enables the stateful caching of session information to specify how asynchronous responses should be handled.
Action: Persistent Message Tracking	Enables session tracking across multiple instances of Forum Sentry gateways
Allowed Destination URLs	Enables whitelist of allowable destinations so as not to provide arbitrary routing control to the calling client
Allowed Reply URLs	Enables whitelist of allowable ReplyTo so as not to provide arbitrary routing control to the calling client

## VALIDATION AND CONFORMANCE TASKS

The tasks under this category are used to inspect and verify target information such as structural conformance.

### TASK: VALIDATE DOCUMENT STRUCTURE (Schema Validation)

The system relies on W3CXML Schemas or DTDs to describe the structure and the rules that govern whether an XML document is valid. During Validation, the system takes a document and maps it to its schema or DTD to enforce the document validity per the schema or DTD. Schemas or DTDs used in this task may be loaded from a File or a URL location.

**TASK LISTS > TASK LIST: MAPPING ATTRIBUTES > TASK: VALIDATE DOCUMENT STRUCTURE**

---

**VALIDATION**

---

Task Type: Validate Document Structure

Task Name\*:

On Error: ☒ Log & Halt Processing ☐ Log & Continue

Error Template:

---

**DOCUMENT VALIDATION**

---

Filename:

Validate against\*: ☒ File

☐ URL

Validate: ☒ Document ☐ Attachments

☐ Automatically load imported files.

**Import**

---

**SELECT ELEMENTS TO VALIDATE**

---

☐ ☒ soap:Envelope

☐ soap:Body

---

Document Elements to Validate

☐ **NAME**

No items to display

**Apply** **Save**

The system supports XSD or DTD Document Structure Validation with standalone schemas, strict or lax. Document Structure Validation with compound schemas (i.e. schemas with include statements and XSD), strict or lax.

The specifications supported on the system for the Validate Document Structure task are:

- W3C XML Schema
- W3C XML 1.0 and 1.1
- Plain Old XML (POX)

## Validate Document Structure Task Screen Terms

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
Error Template	The template policy used to map errors
Filename	Source document to load at design time. If the XSD schema is a complex schema with includes and/or imports, Sentry will prompt for referenced schemas that are dependent.
Validate	The target document for the schema validation. This can be the document itself, or a document send as a MIME, DIME, or MTOM attachment.
Automatically Load Imported Files	Use this option to automatically resolve and URI based include or import references within the scheme. If schema locations are file based, Sentry Web Admin will prompt interactively for required schemas.

### Overview of Validating with a Standalone or Compound Schema

A standalone schema document contains no include statement for additional schemas. A compound schema document imports or includes one or more schemas.

To import a compound schema, you must first upload each referenced schema into the policy. When you attempt to import a compound schema, you will be prompted to select the previously imported included schemas for each import statement in the compound schema.

When the Administrator imports a compound schema, the following events occur:

1. The Administrator is prompted to select an XSD file to import.
2. A message appears, notifying the user that the schema selected is a compound schema.
3. The Administrator loads the compound schema.
4. The Administrator is prompted to select a previously imported schema for each include statement encountered.

The Administrator will repeatedly see the open screen for each new schema referenced in order to select the referenced schema from the list of available schemas (uploaded to the policy).

The Imports and includes text box populates with a read-only listing of the schemas that are associated with the import / include statements in the compound schema.

## TASK: VALIDATE JSON

The Validate JSON task will map a JSON schema to the target document to ensure that the document is valid per the structure and data types specified in the JSON schema.

TASK LISTS > TASK LIST: MAPPING ATTRIBUTES > TASK: VALIDATE JSON

---

**JSON VALIDATION**

---

Task Type: Validate JSON

Task Name\*:

On Error: ☒ Log & Halt Processing ☐ Log & Continue

---

**DATA VALIDATION**

---

Filename:

Validate Against: ☒ File

☐ URL

---

### Validate JSON Task Screen Terms

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
Filename	Source JSON Schema definition document to load at design time.

## TASK: VALIDATE X509 CERTIFICATES

The Validate X509 Certificates task will extract an X509 certificate from within the message in order to authenticate the X509 against the defined Sentry Signer Group to X509 Path Validation.

Note that the Verify Signautre task, the Encrypt Task, and other tasks available on Sentry that are already dependent on X509 processing will have embedded validate X509 capability. This task is not required to be used for those tasks, but rather for more customized processing scenarios such as a custom X509 validation service, would this task be leveraged.

**VALIDATE CERTIFICATES**

Task Type: Validate X.509 Certificates  
Task Name\*:   
On Error: ☒ Log & Halt Processing ☐ Log & Continue  
Signer Group:  [Edit](#)

**SELECT CERTIFICATES TO VALIDATE**

- ☐ soap:Envelope  
    ☐ soap:Body

**Certificates to Validate**

☐ **ELEMENT**

No items to display

[Apply](#)[Save](#)**Validate X509 Certificates Task Screen Terms**

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
Signer Group	The Signer Group policy used to authenticate and validate the X509 using the Sentry DoD PKI Certified X509 Path Validation engine.
Certificates to Validate	The target XPath location of the embedded certificate within the message

## LOGGING AND ARCHIVING TASKS

The tasks under this category are used for alerting, logging or archiving of meta-data information extracted from the request or response.

### TASK: ALERT TASK

This task, when associated with an error template, will send an email alert and/or SNMP trap when the error template is triggered.

Follow the steps below to add an Alert Task.

1. The email settings under the System page will need to be filled in with the appropriate information as seen in the next image:

Forum Sentry > API SECURITY GATEWAY > FORUMSYSTEMS System Name: API Security Gateway A IP Address: 10.5.4.70

**SYSTEM SETTINGS**

NTP Time Server:

Maximum Clock Skew (secs)\*:

Session Timeout (in minutes)\*:

☒ Login Attempts:

SSL Termination Policy\*:  [Edit](#)

SSL Initiation Policy\*:  [Edit](#)

☐ Configuration Database

☐ Block access to unprotected services

☐ Share sessions across policies by cookie name

**EMAIL SETTINGS**

SMTP Mail Server:

SMTP Port:

From email address:

Send system alerts to email address:

[Send Test Email](#)

2. Next, a local user with an email address will need to be added under Access->Users
3. Optionally, SNMP would need to be enabled under Diagnostics->SNMP
4. Add an Alert Task List with the Alert Task:

Forum Sentry > API SECURITY GATEWAY > FORUMSYSTEMS System Name: API Security Gateway A IP Address: 10.5.4.70

**TASK LISTS > TASK LIST: ALERT TASK > TASK: ALERT**

Configuration saved

**ALERT**

Task Type:

Task Name\*:

**CONFIGURATION**

☒ Send email

User\*:  [Edit](#)

Subject\*:

☒ Send SNMP trap

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

5. Create a custom error template and add the Alert Task List as seen next:

**FORUMSENTRY** **API SECURITY GATEWAY** **FORUMSYSTEMS**

**GENERAL**  
 Forum Systems  
 Getting Started  
 Help

**DIAGNOSTICS**

**GATEWAY**

**RESOURCES**

**PKI**  
 Keys  
 Signer Groups  
 CRLs  
 SSH Keys  
 Known Hosts

**Security Policies**  
 OpenPGP  
 SSL  
 Encryption  
 Decryption  
 Signature  
 Verification

**Pattern Match**  
 Pattern Match

**Templates**  
 Error Templates

**Documents**  
 Documents

**WAF**  
 WAF Policies  
 Value Types

**ERROR TEMPLATES > ERROR TEMPLATE DETAILS**

**ERROR TEMPLATE DETAILS**

Template Name\*: Custom-Error-01

Default Error Code\*: 500

Content Type\*: text/xml; charset=utf-8

Processing: Task Lists **Alert Task** [Edit](#)

☐ Disable SOAP fault support (not recommended)

☐ Use MTOM error format for MTOM requests

Default Format:

```
<fs:Error xmlns:fs="http://www.forumsystems.com/2004/04/error">
  <fs:Message>%abortmsg%</fs:Message>
  <fs:SystemName>%sysname%</fs:SystemName>
  <fs:User>%username%</fs:User>
  <fs:Policy>%policy%</fs:Policy>
  <fs:TaskList>%tasklist%</fs:TaskList>
  <fs:Task>%task%</fs:Task>
</fs:Error>
```

SOAP Fault Detail Format:

```
<fs:Detail
xmlns:fs="http://www.forumsystems.com/2004/04/soap-fault-detail">
  <fs:SystemName>%sysname%</fs:SystemName>
  <fs:User>%username%</fs:User>
  <fs:Policy>%policy%</fs:Policy>
  <fs:TaskList>%tasklist%</fs:TaskList>
  <fs:Task>%task%</fs:Task>
</fs:Detail>
```

```
<fs:FaultDetail
xmlns:fs="http://www.forumsystems.com/2004/04/soap-fault-detail">
  <fs:SystemName>%sysname%</fs:SystemName>
  <fs:User>%username%</fs:User>
  <fs:Policy>%policy%</fs:Policy>
  <fs:TaskList>%tasklist%</fs:TaskList>
  <fs:Task>%task%</fs:Task>
```

## Alert Task Screen Terms

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
Send Email	Send an email alert
User	The user with associated email to send alerts to
Subject	The subject the email alerts will have
Send SNMP trap	When checked will send an SNMP trap



## TASK: ARCHIVE DOCUMENT

To perform any Archiving tasks on the product, it is first necessary to configure your archiving database from the Archiving screen. Administrators may extract specific and targeted data for tracking purposes by capturing these elements within a document. This metadata is stored on any JDBC-compliant network database. Administrators build rules that instruct the system which elements to Archive from an intercepted or received document. Once files have been archived, they may be viewed from the Archiving screen.

**ARCHIVE**

Task Type: Archive Document

Task Name\*: Archive Document

On Error: ☒ Log & Halt Processing ☐ Log & Continue

☐ Archive XML document

**SELECT ELEMENTS TO ARCHIVE**

☐ Invoice

- ☒ InvoiceNo
- ☒ OrderDate

☐ Item

- ☐ ProductID

**Document Elements to Archive**

<input type="checkbox"/> ELEMENT	DATA TYPE	COMMENT
<input type="checkbox"/> /Invoice/InvoiceNo	String	ReqInvNum
<input type="checkbox"/> /Invoice/OrderDate	Date	ReqDate

Remove Apply Save

Administrators may archive any of the following:

- an entire Document (that cannot be commented)
- selected Elements (that may be commented)
- both the entire Document and selected Elements

### Archive Document Task Screen Terms

While using the Enrich Message task, please consider the following terms and definitions:

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
Archive XML Document	When checked will archive the entire message
Select Elements to Archive	Xpath expressions that point to the information to be extracted on the inbound transaction to be used for archiving.

## TASK: DISPLAY WSDL URIs

This task is explicitly used by an XML or WSDL (service mode) policy to turn that policy into a WSDL catalog service to display the catalog and meta information from the onboard WSDL policies. No additional settings are required for this task other than simply creating it.

TASK LISTS > TASK LIST: JSON TASKS > TASK: DISPLAY WSDLs URIs

---

**DISPLAY WSDLs URIs**

Task Type: Display WSDLs URIs

Task Name\*:

**Apply**

## Display WSDL URIs Task Screen Terms

No additional settings are required, simply create and associate the task for the behavior to be active.

## TASK: LOG

The Log task is used to induce the logging system to log a message either as specified within the policy, or obtained from the message via XPath query expression.

TASK LISTS > TASK LIST: ENRICH MESSAGE > TASK: LOG

---

**LOG**

Task Type: Log

Task Name\*:

On Error: ☒ Log & Halt Processing ☐ Log & Continue

Logging Level:

Message:

---

**Log Entries**

#	SOURCE TYPE	SOURCE NAME
No items to display		

**Delete** **New**

---

**SELECT ELEMENTS TO LOG**

☐ soap:Envelope

☐ soap:Body

---

**Document Elements to Log**

☐ ELEMENT

No items to display

**Apply** **Save**

## Log Message Screen Terms

While using the Log Message task, please consider the following terms and definitions:

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
Logging Level	Which log level to log the message in the System log
Message	The message to write to the System Log
Select Elements to Log	(optional) XPath expression to extract information from the message to write to the System log.

## TASK: LOG TRANSACTION PROPERTIES

This task, when used will log certain transaction properties to the system log file. These properties include certain User attributes as well as request headers.

The screenshot shows the 'TASK: LOG TRANSACTION PROPERTIES' configuration page in the Forum Sentry API Security Gateway. The interface includes a left sidebar with navigation links like 'GENERAL', 'DIAGNOSTICS', and 'GATEWAY'. The main content area shows the task configuration with fields for 'Task Name' (set to 'Log Transaction Properties'), 'On Error' (radio buttons for 'Log & Halt Processing' and 'Log & Continue'), and 'Logging Level' (a dropdown menu set to 'Error'). There are 'Apply' and 'Save' buttons at the bottom right. The top header displays 'FORUMSENTRY > API SECURITY GATEWAY' and system information like 'System Name: API Security Gateway A' and 'IP Address: 10.5.4.70'.

## Mapping Table Task Screen Terms

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
On Error	Is not used in this Task
Logging Level	Set the log level at which the transaction properties are logged to the system log

## CREDENTIAL GENERATION TASKS

The tasks under this category are used for creating new credentials such as SAML assertions or passwords.

### TASK: GENERATE PASSWORD

This task is used to generate a password value with the given attributes.

TASK LISTS > TASK LIST: SAMPLE 2 > TASK: GENERATE PASSWORD

---

**GENERATE PASSWORD**

Task Type:	Generate Password
Task Name*:	<input type="text" value="Generate Password"/>
Target Type:	<input type="text" value="User Attribute"/>
Target Name*:	<input type="text"/>
Total Length*:	<input type="text" value="8"/>

---

☐ Uppercase characters [A-Z]  
Minimum number of uppercase characters:

☐ Numeric characters [0-9]  
Minimum number of numeric characters:

☐ Special characters  
Minimum number of special characters:   
Characters to be excluded in the password:

### Mapping Table Task Screen Terms

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
Target Type	Location where the newly generated password will be mapped
Target Name	The value name for the password. If the target type is an attribute, this is the attribute name. If the target type is a protocol header, this is the header name.
Total Length	The length of the generated password
Uppercase Characters	Whether to use only uppercase characters
Numeric Characters	Whether to use numeric characters
Special Characters	Whether to use special characters, and defined characters that can be excluded.

## TASK: SAML ASSERTION

The Security Assertions Markup Language (SAML) is an approved standard using the XML protocol for exchanging authentication and authorization credentials over the Web, especially across security boundaries. Combined with XML Signatures, companies can exchange signed SAML assertions that confirm a particular user is authenticated and authorized to access certain network services. The system supports the SAML 1.1 and 2.0 specifications. For more information, refer to <http://www.oasis-open.org>.

As an XML document hops from one destination to another, applying and signing a SAML Assertion at the starting point of the XML document journey eliminates the need for the user to authenticate at each additional hop, as the token added during the Add a SAML Assertion task is passed to all subsequent hops. XML documents arriving at the product may have a SAML Assertion added to it. Additionally, this SAML Assertion may be a User Name token type or an X.509 Binary token type. The product may be configured to generate, as well as sign, the SAML Assertion.

Configuration options available with SAML assertions include:

- Add a SAML Assertion
  - Select Email Identification Format
    - Select Dynamic or Static user to identify
      - Select Authentication Statement Type
      - Select Attribute Statement Type
        - Use Username attribute
        - Use Email attribute
        - Use DN attribute
        - Use Constant attribute
        - Use User attribute (e.g. LDAP)\*
        - Use Cookie attribute
      - Select Authorization Statement Type
  - Select X.509 DN Identification Format
    - Select Dynamic or Static user to identify
      - Select Authentication Statement Type
      - Select Attribute Statement Type
        - Use Username attribute
        - Use Email attribute
        - Use DN attribute
        - Use Constant attribute
        - Use User attribute (e.g. LDAP)\*
        - Use Cookie attribute
      - Select Authorization Statement Type
- Edit a SAML Assertion
- Disable a SAML Assertion
- Remove a SAML Assertion

\* The User attribute is also used for SiteMinder and Tivoli clients.

The specifications supported on the system for the SAML Assertion task are:

- OASIS SAML 1.1
- OASIS SAML 2.0
- WSS Security 1.1
- OASIS WS-Security 2004
- OASIS WS-Security 2004 SAML Token Profile 1.0

### SAML Assertion Task Terms

The following table displays the terms and definitions found in the various screens that are part of the SAML Assertion task:

TERM	DEFINITION
Version	<ul style="list-style-type: none"><li>• With the SAML 1.1 radio button selected, a SAML assertion is generated according to the SAML 1.1 specification.</li><li>• With the SAML 2.0 radio button selected, a SAML assertion is generated according to the SAML 2.0 specification.</li></ul>
Issuer	Specifies the issuer of the assertion. The issuer name should match an issuer name allowed by the recipient if the recipient performs issuer checking. (Issuer checking is optional in the Identity Document task.)
Include a validity start time and Time to start	With this field checked, enter the time at which the assertion becomes valid.
Assertion expires and Time to expire	Specifies an optimal time limit when an assertion expires and becomes invalid. If a SAML attribute assertion is configured to include a session cookie that expires, and the assertion itself does not have a different expiration configured, the generated SAML assertion is set by the system to expire when the cookie expires.
Disallow caching of this assertion	<ul style="list-style-type: none"><li>• When checked, specifies that the assertion is to be used by the recipient one time only and should not be cached for later use.</li><li>• When unchecked, allows caching, which may decrease security.</li></ul>
Disallow reuse of this assertion	<ul style="list-style-type: none"><li>• When checked, enables SAML replay detection and only allows an assertion to be used once.</li><li>• When unchecked, the SAML Assertion may be used more than once. SAML replay detection is disabled.</li></ul>
Identification Format	<ul style="list-style-type: none"><li>• With Email checked, identifies SAML Email token. The email identification format supports local and LDAP users only.</li><li>• With X.509 Distinguished Name checked, identifies SAML X.509 DN token.</li></ul>

TERM	DEFINITION
Include the identifier format URI	Explicitly specify the format of the name identifier (e.g. email or X.509) in the assertion. Including the format may help a recipient in processing the assertion if the recipient does not already know which format to expect.
Dynamic, based on established identity	When selected, applies the email or X.509 Distinguished Name of the user identified earlier during the User Identity and Access Control task.
Static, based on a specified user	When selected, applies the Email of the selected user or the subject DN of the selected X.509 certificate to this SAML Assertion.
Statement Type	<p>Specifies the statement type of SAML assertion to generate. Statement types are not mutually exclusive.</p> <ul style="list-style-type: none"> <li>• Authentication - Asserts that the user is authenticated and records the type of authentication used.</li> <li>• Attribute - Associates specified attributes with the user.</li> <li>• Authorization - Grants / denies the user access to a specified resource.</li> </ul>
Include the client IP address	This option includes the IP address of the authenticated client in the SAML authentication statement.
Signature Policy	The XML Signature Policy name to use for signing.
Include certificates	When checked, includes the X.509 certificate(s) when signing.
Attribute Namespace	This mandatory field specifies the namespace URI of the SAML attribute.
Attribute Name	This mandatory field specifies the name of the SAML attribute.
Attribute Value Type	<ul style="list-style-type: none"> <li>• With Username selected, an attribute with the value of the user name is included in the assertion.</li> <li>• With Email selected, an attribute with the value of the user email address is included in the assertion.</li> <li>• With DN selected, an attribute with the value of the user DN is included in the assertion.</li> <li>• With Constant selected, an attribute with the specified constant value is included in the assertion. The Constant field accepts any keyboard character, from 1-256 characters in length.</li> <li>• With User attribute selected, the specified user attribute is obtained from LDAP or an identity server and included in the assertion. Multiple attribute names may be entered comma delimited. This functionality can also be used for SiteMinder and Tivoli.</li> <li>• With Cookie selected, the specified cookie is included in the assertion. This can be used for any type of cookie, e.g., a standard HTTP cookie.</li> </ul>

TERM	DEFINITION
Authorization Resource	This mandatory field specifies the URI of the authorized resource. Leaving the field blank equates to the URI being an empty string, which is defined to identify the current document.
Authorization Namespace	This mandatory field specifies the namespace URI of the authorized action.
Authorization Action	<p>This mandatory field specifies the authorized action, which depends on the value that the Administrator first types in for the action namespace. If the Administrator uses the default action namespace that appears on the screen, “urn:oasis:names:tc:SAML:1.0:action:rwdc-negation”, then the user could type in one or more of the following values for the action:</p> <ul style="list-style-type: none"> <li>• Read - The subject may read the resource.</li> <li>• Write - The subject may modify the resource.</li> <li>• Execute - The subject may execute the resource.</li> <li>• Delete - The subject may delete the resource.</li> <li>• Control - The subject may specify the access control policy for the resource.</li> </ul> <p>Actions prefixed with a tilde (~) are negated permissions and are used to affirmatively specify that the stated permission is denied. Thus, a subject described as being authorized to perform the action ~Read is affirmatively denied read permission.</p> <p>A SAML authority MUST NOT authorize both an action and its negated form.</p>

## TASK- WS-SECURITY HEADER

The WS-Security Header task allows users to add a WS-Security header to incoming XML documents. The WS-Security Header, contained in a SOAP message along with the body of the XML document, includes a variety of information about the XML document, such as:

- Who originally generated the XML document.
- Who authenticated the person who generated the XML document.
- Which elements in the XML document are signed and/or encrypted, and by whom.
- Who authenticated the Signer and/or Encrypter of this XML document.
- Which Signatures are included in this XML document.
- Who was the CA for all included Signatures.
- What is the START destination or first hop for this XML document.
- What are intermediary or subsequent hops for this XML document, and in what sequence.
- What is the END destination or last hop for this XML document.
- What is the nature of this XML document.
- What is a summary of the contents of this XML document.

Configuration options available with WS-Security Headers include:

- Add a WS-Security Header
  - Select No token
  - Select Username token
    - Select Dynamic or Static user to identify
      - Select No Password type



- Select Clear Text Password type
  - Select SHA1 Digest Password type
- Select X.509 binary token
  - Select Dynamic or Static user to identify
- Select SAML token
  - Select SAML Email or X.509 SAML ID format
    - Select Dynamic user to identify
    - Select Static user to identify user
      - Select Authentication SAML Statement Type
        - Use No token for Security token authentication
        - Use Username token for Security token authentication
        - Use X.509 binary token for Security token authentication
        - Use SAML token for Security token authentication
      - Select Attribute SAML Statement Type
        - Use Username attribute
        - Use Email attribute
        - Use DN attribute
        - Use Constant attribute
        - Use User attribute (e.g. LDAP)
      - Select Authorization SAML Statement Type
- Edit WS-Security Header
- Disable WS-Security Header
- Remove WS-Security Header

The specifications supported on the system for the WS-Security Header task are:

- OASIS WS-Security 1.1
- OASIS WS-Security 2004
- OASIS WS-Security 2004 SAML Token Profile 1.0
- OASIS WS-Security 2004 Username Token Profile 1.0
- OASIS WS-Security 2004 X.509 Certificate Token Profile 1.0
- OASIS SAML 1.1

### Prerequisites for All WS-Security Header Tasks

Before performing any of these operations listed above, except for No access control, it is assumed that:

- at minimum, one User, Group and ACL have been created in the User, Group and ACL Management screens.
- this user has been assigned membership into a Group (from the USER DETAILS screen or from the GROUP DETAILS screen), and the Group has been assigned membership into the ACL from the ACL DETAILS screen.

**Note:** All operations performed in this chapter are performed statically. To perform these operations dynamically requires the User Identity and Access Control task.

An example of dynamically configuring a token is presented later in this document under Add User Identity/Access Control by WS-Security Header with User Name Token.

### Replay Verification with WS-Security Header Tasks

Replay Verification is available on WS-Security Header Username tokens, and is automatic when a nonce is received. Checking the **Include a nonce** option will allow a message to be received once, but not more than once. A nonce is only valid for five minutes.

**Note:** Because Replay Verification is time-sensitive, you may relax the time set for the time zone of your Client and Server using the Maximum Clock Skew (in seconds) option on the **System->Settings>System** screen.

### WS-Security Header Task Wizard Terms

The following table displays all the terms and definitions found in the WS Security Header task wizard:

TERM	DEFINITION
Task Name	The name given to this task. Users may accept the default task name or give the task a unique name.
WS-Security processing by recipient is mandatory	<ul style="list-style-type: none"><li>• When checked, WS-Security processing is mandatory</li></ul>
Must Understand checkbox	<ul style="list-style-type: none"><li>• When checked, makes the recipient processing of the WS-Security SOAP header mandatory so that web services which receive the message must be WS-Security-aware.</li><li>• When unchecked, WS-Security processing by the recipient is not mandatory.</li></ul>
Time to Live	The Time to live may have 1 to 20 numeric characters. The default time to expire is 1 minute.

Security Token Type	<ul style="list-style-type: none"> <li>• With No Token selected, no security token is generated.</li> <li>• With Username token selected, a Username token is added to the document.</li> <li>• With X.509 binary token selected, an X.509 binary token is added to the document.</li> <li>• With SAML token selected, a SAML token is added to the document.</li> </ul>
Password type	<ul style="list-style-type: none"> <li>• With None selected, no password is selected.</li> <li>• With Clear Text selected, a password is included in clear text format.</li> <li>• With SHA 1 Digest selected, a SHA 1 digest of the password is included.</li> </ul>
Include Nonce	When checked, generates a nonce for each username token. The nonce secures SHA 1 password digests and enables replay detection. Replay detection in the system is automatic when a nonce is received. A nonce is only valid for five minutes.
<b>TERM</b>	<b>DEFINITION</b>
Include timestamp	When checked, generates a timestamp for each username token. The timestamp secures SHA 1 password digests and enables replay detection. Replay detection in the system is automatic when a username token timestamp is received. Username tokens with timestamps are only valid for five minutes.
X.509 Identification	<ul style="list-style-type: none"> <li>• Selecting the Dynamic, based on protocol certificate radio button adds the X.509 certificate of the run-time client or user to this WS-Security Header.</li> <li>• Selecting the Static, based on specified user radio button adds the X.509 certificate of the specified user to this WS-Security Header.</li> </ul>
Sign SAML Assertion	When checked, applies the Signature Policy selected in the Signature Policy drop down list to this task.
Signature Policy	The Signature Policy selected from the drop down list to be applied to the SAML Assertion in this task.
Include Certificates	When checked, includes the X.509 certificate(s) when signing.

## SECURITY PROCESSING TASKS

The tasks under this category are used for applying security and cryptography tasks more granularly to target data and attributes. This includes XML or JSON security processing, selective virus scanning, and regular expression pattern matching.

### TASK: DECRYPT ELEMENTS

The Decrypt Elements task may be used to decrypt some or all encrypted portions of XML documents and attachments. The Decrypt Elements task uses the private key specified in a Decryption policy to perform decryption and can enforce the use of specified encryption algorithms.

TASK LISTS > TASK LIST: REPLACEPINGRESPONSE > TASK: DECRYPT ELEMENTS

**DECRYPT**

Task Type: Decrypt Elements

Task Name\*: Decrypt Elements

On Error: ☒ Log & Halt Processing ☐ Log & Continue

**DECRYPTION POLICIES**

Decryption policy: ▼

☒ **SELECT ELEMENTS TO DECRYPT**

☐ soap:Envelope

☐ soap:Body

☐ ccn:Ping

☐ ccn:Input

**Elements to Decrypt**

☐ **PATH**

No items to display

Apply Save

The specifications supported on the system for the Decrypt Element task are:

- W3C XML Encryption Syntax and Processing
- OASIS WS-Security 1.1
- OASIS WS-Security SOAP Messages with Attachments Profile 1.1
- OASIS WS-Security 2004
- OASIS SAML 2.0

### Element-Level and Content-Level Decryption

When you decrypt an element or content, you are reversing the encryption and restoring the document to its original structure. Decryption may be required in order to further process the document. When both decryption and schema validation tasks are used, decryption is usually appropriate before the Validation task. In the instructions presented in this document, you will be decrypting before validating the Incoming Document.

### Decryption Screen Terms

While decrypting a document, please consider the following terms and definitions:

TERM	DESCRIPTION OF OPTIONS
------	------------------------

<b>DECRYPT</b>	
On Error	<ul style="list-style-type: none"> <li>• With Log &amp; Halt Processing selected, if an error is encountered, the decryption process will log an error and halt processing.</li> <li>• With Log &amp; Continue selected, if an error is encountered, the decryption process will log an error and continue processing.</li> </ul>
<b>DECYPTION PROPERTIES</b>	
Decryption policy	A listing of current Decryption Policies to select from.
<b>ELEMENTS TO DECRYPT</b>	
Path	Node selected for decrypt options.

## TASK: ENCRYPT ELEMENTS

This task allows the encryption of granular information of the node or node content using XPath to selectively target the specific node, element, or attribute to encrypt.

TASK LISTS > TASK LIST: REPLACEINGRESPONSE > TASK: ENCRYPT ELEMENTS

No encryption policies found. Please create a new encryption policy before proceeding.

### ENCRYPT

Task Type:	Encrypt Elements
Task Name*:	<input type="text" value="EncryptElements"/>
On Error:	<input checked="" type="radio"/> Log & Halt Processing <input type="radio"/> Log & Continue

### ENCRYPTION PROPERTIES

Type:	<input checked="" type="radio"/> Encrypt Element <input type="radio"/> Encrypt Content
Method:	<input checked="" type="radio"/> WSS 1.1 <input type="radio"/> WSS 2004 <input type="radio"/> XML Encryption
Encryption policy:	<input type="text" value=""/>
Key Identifier:	<input checked="" type="radio"/> SerialNumber <input type="radio"/> X.509 <input type="radio"/> SubjectKeyIdentifier <input type="radio"/> ThumbprintSHA1 <input type="radio"/> Subject
<input type="checkbox"/> Encrypt attachments	
<input type="checkbox"/> Canonicalize base64Binary data (MTOM-compatible)	

### SELECT ELEMENTS TO ENCRYPT

<input type="checkbox"/> soap:Envelope
<input type="checkbox"/> soap:Body
<input type="checkbox"/> ccn:Ping
<input type="checkbox"/> ccn:Input

### Elements to Encrypt

<input type="checkbox"/> PATH
No items to display

The Encrypt Elements task may be used to encrypt some or all encrypted portions of XML documents and attachments to ensure confidentiality. The Encrypt Elements task uses the public key specified in an Encryption policy to perform encryption with a specified encryption algorithm.

When used in conjunction with digital signatures, encryption can precede or follow the signature task, as necessary. Multiple encryption tasks may use the same or different public keys.

The specifications supported on the system for the Encrypt Elements task are:

- W3C XML Encryption Syntax and Processing
- OASIS WS-Security 1.1
- OASIS WS-Security SOAP Messages with Attachments Profile 1.1
- OASIS WS-Security 2004
- OASIS SAML 2.0

## Element-Level and Content-Level Encryption

When you encrypt an element or content, you usually render the document inconsistent with the document schema because the XML schema you use no longer matches the changed structure. The encrypted element or content is replaced by an **EncryptedData** element. When both encryption and schema validation tasks are used, encryption is usually performed appropriate after the Validation task.

## Encrypting Attachments

You may also add an encryption policy to the attachments present in a SOAP with Attachments message. You do not need a special policy to work with SOAP attachments in the product.

At runtime, the system must have the appropriate request filter(s) configured for the XML or WSDL policy in order to receive attachments. For more information, refer to the “Attachments request filter” section in *Forum Systems Sentry™ Version 9 XML Policies Guide*.

### **Key Identifier**

The Encrypt Elements task includes a choice of four types of key identifiers:

- SerialNumber, which uses the X.509 issuer DN and serial number.
- X.509, which uses the complete X.509.
- SubjectKeyIdentifier, which uses the X.509 v3 SubjectKeyIdentifier extension.
- Subject, which uses the X.509 subject DN.

### **Encryption Method**

Encryption options are:

- XML Encryption
- WSS 2004
- WSS 1.1

### **XML Encryption Method**

The XML Encryption Method allows Administrators to specify that the encrypted symmetric key (i.e. the EncryptedKey element) used for the element or content encryption will be located within the encrypted element (i.e. the EncryptedData element). This method is compliant with the XML Encryption Syntax and Processing specification (<http://www.w3.org/TR/xmlenc-core/>).

### **WS-Security Specification**

The WS-Security specification allows Administrators to specify that the encrypted symmetric key used for the encryption will be located in a WS-Security header in accordance with the WSS 2004 or WSS 1.1 specification.

## Encryption Screen Terms

While encrypting a document, please consider the following terms and definitions:

TERM	DESCRIPTION OF OPTIONS
<b>ENCRYPT</b>	
On Error	<ul style="list-style-type: none"><li>• With Log &amp; Halt Processing selected, if an error is encountered, the encryption process will log an error and halt processing.</li><li>• With Log &amp; Continue selected, if an error is encountered, the encryption process will log an error and continue processing.</li></ul>
<b>ENCRYPTION PROPERTIES</b>	
Type	<ul style="list-style-type: none"><li>• With Encrypt Element selected, the encryption policy is applied to the entire node selected.</li><li>• With Encrypt Content selected, the encryption policy is applied to the content of the node selected.</li></ul>
Method	<ul style="list-style-type: none"><li>• With WSS 1.1 selected, applies the WSS 1.1 WS-Security specification. The encrypted symmetric key used for the encryption will be placed in a WS-Security header in accordance with the OASIS WS-Security 1.1 specification</li><li>• With WSS 2004 selected, applies the WSS 2004 WS-Security specification. The encrypted symmetric key used for the encryption will be placed in a WS-Security header in accordance with the WS-Security 2004 specification.</li><li>• With XML Encryption selected, applies the XML Encryption specification. The encrypted symmetric key used for the element or content encryption will be stored in place with the encrypted element.</li></ul>
Encryption policy	The Encryption policy to use.
Encrypt attachments	When checked, any SOAP attachments present in the message will be encrypted by the product.
Key Identifier	<ul style="list-style-type: none"><li>• With SerialNumber selected, the X.509 issuer DN and serial number is used for identifying the key.</li><li>• With X.509 selected, the complete X.509 is used for identifying the key.</li><li>• With SubjectKeyIdentifier selected, the X.509 v3 SubjectKeyIdentifier extension is used for identifying the key. Although a Key Identifier may be selected, the WSS 2004 specification prefers the SerialNumber option.</li><li>• With Subject selected, the X.509 subject DN is used for identifying the key.</li></ul>
<b>ELEMENTS TO ENCRYPT</b>	
Path	Node selected for encrypt options.



## TASK: JWE ENCRYPTION

This task encrypts data using JSON Web Encryption (JWE). It supports various algorithms, encryption algorithms, compression, custom attributes, media types, and key IDs. The output format can be JSON or an attribute.

TASK LISTS > TASK LIST: SAMPLE 1 > TASK: JWE ENCRYPTION

JWE ENCRYPTION	
Task Type:	JWE Encryption
Task Name*:	<input type="text" value="JWE Encryption"/>
Encryption Policy*:	<input type="button" value="v"/>
Algorithm*:	<input type="button" value="RSA using OAEP-256"/>
Encryption Algorithm*:	<input type="button" value="AES GCM (256-bit)"/>
JWE Serialization:	<input checked="" type="radio"/> Compact Serialization: <input type="radio"/> JSON Serialization (Not Supported):
Media Type:	<input type="text" value="JOSE"/>
Key ID:	<input type="text"/>
Compression Algorithm:	<input type="button" value="Uncompressed"/>
Additional Headers:	<div><input type="text" value="iat,exp"/></div>
Output Format:	<input type="button" value="JSON"/>
JSON Key*:	<input type="text"/>
JWE HEADERS	
alg:	RSA-OAEP-256
enc:	A256GCM
typ	
kid	
zip	

## JWE Encryption Screen Terms

TERM	DESCRIPTION OF OPTIONS
Encryption Policy	The encryption policy defined under Resources->Security Policies->Encryption
Algorithm	<p>The cryptographic algorithms used to encrypt and decrypt the JWE content. These algorithms are independent of the algorithms specified in the encryption policy, but depending on the key type, they need to be compatible. List of supported algorithms:</p> <ul style="list-style-type: none"><li>• RSA using OAEP</li><li>• RSA using OAEP-256</li><li>• RSA (v1.5 padding)</li><li>• AES Key Wrap (128-bit)</li><li>• AES Key Wrap (192-bit)</li><li>• AES Key Wrap (256-bit)</li></ul>

	<ul style="list-style-type: none"> <li>• AES GCM Key Wrap (128-bit)</li> <li>• AES GCM Key Wrap (192-bit)</li> <li>• AES GCM Key Wrap (256-bit)</li> <li>• PBE S2 with HMAC SHA-256 (128-bit)</li> <li>• PBE S2 with HMAC SHA-384 (192-bit)</li> <li>• PBE S2 with HMAC SHA-512 (256-bit)</li> <li>• ECDH with AES Key Wrap (128-bit) (Not currently supported by Encryption Policy)</li> <li>• ECDH with AES Key Wrap (192-bit) (Not currently supported by Encryption Policy)</li> <li>• ECDH with AES Key Wrap (256-bit) (Not currently supported by Encryption Policy)</li> <li>• ECDH Direct (Not currently supported by Encryption Policy)</li> <li>• Direct encryption</li> </ul>
Encryption Algorithm	<p>The algorithm used to encrypt the JWE content. This algorithm should match the bit size of the symmetric key used in the encryption policy if a symmetric key is employed. List of supported algorithms:</p> <ul style="list-style-type: none"> <li>• AES GCM (128-bit)</li> <li>• AES GCM (192-bit)</li> <li>• AES GCM (256-bit)</li> <li>• AES CBC (128-bit) with HMAC SHA-256</li> <li>• AES CBC (192-bit) with HMAC SHA-384</li> <li>• AES CBC (256-bit) with HMAC SHA-512</li> </ul>
JWE Serialization	The format used to represent the JWE data structure. JWE compact serialization is the only serialization currently supported.
Media Type	The media type attribute in the JWE. (typ)
Key Id	A unique identifier for the key used for encryption. (kid)
Compression Algorithm	The algorithm used to compress the JWE content before encryption (optional). (zip)
Additional Headers	A comma-separated list of custom attributes (key-value pairs) that can be included in the JWE header.
Output Format	<p>The format in which the encrypted data is returned. Supported formats:</p> <ul style="list-style-type: none"> <li>• JSON</li> <li>• Attribute</li> <li>• Document</li> </ul>

## Encryption Policy and JWE Algorithms

It's important to note that the algorithms specified in the encryption policy are not directly used for JWE encryption. However, they determine the type of key used for encryption, and depending on the key type, they need to be compatible with the chosen JWE algorithm:

- **RSA Key:** If the JWE Encryption is using an RSA key, the encryption policy should provide an RSA key. The JWE algorithm should be compatible with RSA public-key encryption.
- **Symmetric Key:** For symmetric keys, the encryption policy should specify a key with the appropriate bit size. The encryption algorithm (e.g., AES GCM) should use a key size that matches the bit size of the symmetric key provided by the encryption policy

## Input and Output Format Details

- **JSON:**
  - Creates a JSON document containing the signed data.
  - Requires configuration of the JSON Key field to specify where the output is stored.
  - Example output:

```
{"jws": "<jws>"}
```

(where <jws> is the base64-encoded signature)

- **Attribute:**
  - Stores the signed data in the configured Attribute Name.
- **Document:**
  - Stores the signed data directly within the running document. No additional configuration is needed.

## Attribute Name and JSON Key

- These fields are only displayed and required when their corresponding Output Format (Attribute or JSON) is selected, respectively.

## Signature Policy

- The Signature Policy determines the key used for signing.
- For RSA keys, the key size is not relevant.
- For Elliptic Curve (EC) keys and HS\* algorithms, the key parameters (curve type or hash function, respectively) must match the chosen Signature Algorithm.

## Example Configuration

The following is an example audit log entry that demonstrates a JWE Encryption configuration:  
Succeeded - Task: 'JWE Encryption' Type:

Task Name: JWE Encryption

Created: 2024/04/16 12:16

Lock: Disabled

Enabled: Yes

Task Type: JWE Encryption

Encryption Policy: Encryption\_Policy

Algorithm: RSA using OAEP-256

Encryption Algorithm: AES GCM (256-bit)

JWE Serialization: JWE Compact Serialization  
Media Type:  
Key ID: 0cdde45d-5bc5-4d48-b1c1-c4fc1a4b0344  
Compression Algorithm: Uncompressed  
Output Format: JSON  
JSON Key: encData  
Additional Headers: iat

## TASK: JWE DECRYPTION

This task decrypts data using JSON Web Encryption (JWE). It supports various algorithms, encryption algorithms, compression, custom attributes, media types, and key IDs. The output format can be JSON or an attribute.

TASK LISTS > TASK LIST: NEW TASK LIST > TASK: JWE DECRYPTION

**JWE DECRYPTION**

Task Type:

JWE Decryption

Task Name:\*

Decryption Policy:\*

Decryption\_Policy\_TOTP\_Secret ▾

[Edit](#)

Input Format:

JSON ▾

JSON Key:\*

Algorithm Constraints:

[None] ▾

Algorithms:

Encryption Algorithm Constraints:

[None] ▾

Encryption Algorithms:

Apply

Save

## JWE Decryption Screen Terms

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
Decryption Policy	The task utilizes the configured decryption policy to obtain the appropriate key for decryption
Input Format	Determines the format of the encrypted data. Choices include: <ul style="list-style-type: none"><li>• JSON</li><li>• Attribute</li><li>• Document</li></ul>
JSON Key	<p>This determines where to retrieve the JWE Token</p> <p><b>JWE Retrieval:</b> The task retrieves the JWE token from the input document based on the configured format:</p> <ul style="list-style-type: none"><li>• <b>Attribute-based:</b> Reads the JWE token from the specified Attribute Name attribute within the session.</li><li>• <b>Document-based:</b> Reads the JWE token from the JSON key specified using the document</li></ul>
Algorithm Constraints	Choose the algorithm constraint type:

	<ul style="list-style-type: none"> <li>• None: No constraints are applied to the algorithms.</li> <li>• Permit: Specify a list of algorithms to permit.</li> <li>• Block: Specify a list of algorithms to block.</li> </ul>
Algorithms	The Algorithms to constrain
Encryption Algorithm Constraints	Choose the algorithm constraint type: <ul style="list-style-type: none"> <li>• None: No constraints are applied to the algorithms.</li> <li>• Permit: Specify a list of algorithms to permit.</li> </ul>
	Block: Specify a list of algorithms to block.
Encryption Algorithms	The Encryption Algorithms to constrain

## Document Handling

1. **JWE Retrieval:** The task retrieves the JWE token from the input document based on the configured format:
  - **Attribute-based:** Reads the JWE token from the specified Attribute Name attribute within the session.
  - **Document-based:** Reads the JWE token from the JSON key specified using the document
2. **Header Parsing:** The task parses the headers of the retrieved JWE token to identify the used algorithms.
3. **Algorithm Constraints Check:** The task verifies if the identified algorithms are allowed based on any configured constraints.
4. **Decryption Policy:** The task utilizes the configured decryption policy to obtain the appropriate private key for decryption.
5. **Decryption:** The JWE token is decrypted using the retrieved private key.
6. **Clear Text Storage:** The decrypted clear text value is stored back into the document.

TASK: JWS SIGNATURE

This task provides the ability to apply a JWS Signature

TASK LISTS > TASK LIST: NEW TASK LIST > TASK: JWS SIGNATURE

JWS SIGNATURE

Task Type:

JWS Signature

Task Name:\*

JWS Signature

Signature Policy:\*

Input Format:

Document

Signature Algorithm:\*

HS256

Media Type:

JOSE

Key ID:

Additional Headers:

iat,exp

Output Format:

Document

Apply

Save

JWS Signature Screen Terms

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
Signature Policy	Specifies the policy used to obtain the appropriate key for signing
Input Format	Defines the format of the data to be signed. Choices include: <ul style="list-style-type: none"><li>JSON</li><li>Attribute</li><li>Document</li></ul>
Signature Algorithm	Selects the signing algorithm. Supported algorithms are: <ul style="list-style-type: none"><li>HMAC using SHA-2<ul style="list-style-type: none"><li>HS256</li><li>HS384</li><li>HS512</li></ul></li><li>RSASSA-PKCS1-V1_5 Digital Signatures with SHA-2<ul style="list-style-type: none"><li>RS256</li><li>RS384</li><li>RS512</li></ul></li><li>RSASSA-PSS Digital Signatures with SHA-2<ul style="list-style-type: none"><li>PS256</li><li>PS384</li><li>PS512</li></ul></li></ul>

---

	<ul style="list-style-type: none"> <li>• PS512 RSA PSS using SHA512</li> </ul>
Media Type	(Optional) Specifies the media type of the signed data
Key ID	(Optional) Used to identify the specific key within the Signature Policy
Additional Headers	(Optional) Allows for inclusion of additional custom headers in the JWS object
Output Format	Determines the format of the signed data. Choices include: <ul style="list-style-type: none"> <li>• JSON</li> <li>• Attribute</li> <li>• Document</li> </ul>

---

#### Input and Output Format Details:

- **JSON:**
  - Creates a JSON document containing the signed data.
  - Requires configuration of the JSON Key field to specify where the output is stored.
  - Example output:

```
{"jws": "<jws>"}
```

(where <jws> is the base64-encoded signature)

- **Attribute:**
  - Stores the signed data in the configured Attribute Name.
- **Document:**
  - Stores the signed data directly within the running document. No additional configuration is needed.

#### Attribute Name and JSON Key:

- These fields are only displayed and required when their corresponding Output Format (Attribute or JSON) is selected, respectively.



## TASK: JWS VERIFICATION

This task provides the ability to verify a JWS Signature

TASK LISTS > TASK LIST: NEW TASK LIST > TASK: JWS VERIFICATION

**JWS VERIFICATION**

Task Type:

JWS Verification

Task Name:\*

JWS Verification

Key Source:

Verification Policy ▾

Verification Policy:\*

▾

Algorithm Constraints:

[None] ▾

Algorithms:

Apply

Save

### JWS Verification Screen Terms

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
Key Source	<ul style="list-style-type: none"><li>• Json Web Key</li><li>• Verification Policy</li></ul>
Verification Policy	Specify either the <b>Verification Policy</b> or the <b>JWK Attribute Name</b> for key retrieval
Algorithm	Select the algorithms relevant to the constraint type. The available options are: <ul style="list-style-type: none"><li>• HS256 HMAC using SHA256</li><li>• HS384 HMAC using SHA384</li><li>• HS512 HMAC using SHA512</li><li>• RS256 RSA using SHA256</li><li>• RS384 RSA using SHA384</li><li>• RS512 RSA using SHA512</li><li>• PS256 RSA PSS using SHA256</li><li>• PS384 RSA PSS using SHA384</li><li>• PS512 RSA PSS using SHA512</li><li>• ES256 ECDSA using P-256 Curve and SHA256</li><li>• ES384 ECDSA using P-256 Curve and SHA256</li><li>• ES512 RSA PSS using SHA512</li></ul>
Algorithm Constraints	Choose the algorithm constraint type: <ul style="list-style-type: none"><li>• None: No constraints are applied to the algorithms.</li></ul>

- Permit: Specify a list of algorithms to permit.
- Block: Specify a list of algorithms to block.

## Document Handling

- The **signed document** is retrieved from the input document.
- The **verified document** (with the signature removed) is placed back into the output document for further processing or use.

Ensure all required parameters are configured correctly for a successful JWS Verification task execution.

## TASK: PATTERN MATCH

The Pattern Match task is used to invoke defined Pattern Match policies against the target document being processed.

TASK LISTS > TASK LIST: PATTERNMATCH > TASK: PATTERN MATCH

---

**PATTERN MATCH**

Task Type: Pattern Match

Task Name\*:

☐ Trigger pattern match IDP rule on violation

**Match Policies**

#	TYPE	NAME	REQUIRED	POLICY NAME	STATUS
1	XML Element Content			Credit_Card_Number	●
2	Protocol Header	TEST	✓	MS_SQL_Injection	●

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

---

**SELECT ELEMENTS**

☐ soap:Envelope
 

- ☐ soap:Body

**Element Match Policies**

ELEMENT	POLICY NAME
No items to display	

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

## Pattern Match Task Screen Terms

While using the Pattern Match task, please consider the following terms and definitions:

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
Match Policies	Rules and criteria to associate a RegEx Pattern Match Task to a target value
Select Elements	When the pattern match target is an XML element, this allows XPath expression targets to the elements/attributes that are to be evaluated.

Pattern Matching target options include:

- **XML Element Content**
  - The targeted node value or attribute value from the select elements element match policy
- **Protocol Header**
  - If the task is operating on the request, this represents the request header. If the task is operating on the response, this setting represents the response header
- **Request Header**
  - The header from the inbound request from the client
- **Response Header**
  - The header from the response from the back-end system
- **User Attribute**
  - A general attribute type the can be referenced by other tasks
- **Query Parameter**
  - A target name value pair to add to the URI for the back-end server request from Sentry
- **Template**
  - A variable that can be referenced within custom text or templates
- **Cookie**
  - Using this setting will result in a SET-COOKIE header being created with the cookie value as the value specified.
- **HTTP Method**
  - GET or POST
- **Request Path**
  - The request path of the inbound client request, not including query parameters
- **HTTP Status Code**
  - The response code from the back-end system response back to Sentry

## PATTERN MATCH POLICIES

These are the policies that define the regular expressions. Once defined, these policies are consumed by the Pattern Match task.



### Pattern Match Policy Screen Terms

While using the Pattern Match Policy, please consider the following terms and definitions:

TERM	DESCRIPTION OF OPTIONS
Policy Name	The name of the policy
Mode	Allow or Deny based on the RegEx match

Regular Expression	RegEx Pattern to match against
Replacement	Replacement value for all the matches
Replace	Check if a replacement value is used

TASK LISTS > TASK LIST: PATTERNMATCH > TASK: PATTERN MATCH

#### PATTERN MATCH

Task Type: Pattern Match

Task Name\*:

☐ Trigger pattern match IDP rule on violation

#### Match Policies

#	TYPE	NAME	REQUIRED	POLICY NAME	STATUS
1	XML Element Content			Credit_Card_Number	<span style="color: green;">●</span>
2	Protocol Header	TEST	✓	MS_SQL_Injection	<span style="color: green;">●</span>

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

#### SELECT ELEMENTS

- ☐ soap:Envelope
  - ☐ soap:Body

#### Element Match Policies

ELEMENT	POLICY NAME
No items to display	

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

### Pattern Match Control Flow

While using the Pattern Match task, the flow of execution can be controlled based on the ALLOW or DENY setting on the Pattern Policy itself and the checkbox for “Trigger Pattern Match IDP Rule on Violation”. Violation means that the pattern is triggered and the ALLOW/DENY is enforced by ensuring that the IDP rule Pattern Match Policy Violation is set.

## IDP RULE POLICIES > IDP RULE DETAILS

### DETECTION SETTINGS

IDP Rule Name\*: IDP\_Rule\_Pattern\_Match\_Violation  
Description:  
Criterion: Pattern match policy violation ▼

### THRESHOLD

Value: 0 KB ▼  
Period: Second ▼

### ENFORCEMENT SETTINGS

☐ Enforce only on user group: AdminRoleWSDLOnly ▼ [Edit](#)  
☐ Enforce by IP  
☐ Enforce by user

### IDP ACTION

IDP Action: Abort ▼ [Edit](#)  
Abort Message:

### IDP SCHEDULE

IDP Schedule: Anytime ▼ [Edit](#)

[Create](#)

## TASK: RECEIVE SIGNATURE CONFIRMATION

The Receive Signature Confirmation task is used in response processing by a document sender to confirm receipt by the recipient of any signatures sent in the outgoing request document.

### TASK LISTS > TASK LIST: NEW TASK LIST > TASK: RECEIVE SIGNATURE CONFIRMATION

#### RECEIVE SIGNATURE CONFIRMATION

Task Type: Receive Signature Confirmation  
Task Name\*: Receive Signature Confirmation

## Receive Signature ConfirmationTask Screen Terms

No additional settings are required, simply create and associate the task for the behavior to be active.

## TASK: SEND SIGNATURE CONFIRMATION

The Send Signature Confirmation task is used in response processing by a document recipient to confirm receipt of any signatures received in the incoming request document.

### TASK LISTS > TASK LIST: NEW TASK LIST > TASK: RECEIVE SIGNATURE CONFIRMATION

#### RECEIVE SIGNATURE CONFIRMATION

Task Type: Receive Signature Confirmation  
Task Name\*: Receive Signature Confirmation

## Send Signature ConfirmationTask Screen Terms

No additional settings are required, simply create and associate the task for the behavior to be active.

## TASK: SIGN DOCUMENT

The Sign Document task provides a means of adding digital signatures to a document to ensure integrity and support authentication and non-repudiation. A digital signature may cover one, multiple, or all portions of an XML document and attachments. The Sign Document task uses the private key specified in a Signature Policy to sign using specified algorithms. An option is also provided

The specifications supported on the system for the Sign Document task are:

- W3C XML-Signature Syntax and Processing
- W3C Canonical XML Version 1.0
- W3C Exclusive XML Canonicalization Version 1.0
- OASIS WS-Security 1.1
- OASIS WS-Security SOAP Messages with Attachments Profile 1.1
- OASIS WS-Security 2004
- OASIS ebXML Message Service 2.0

## Signature Types Supported

The signature types supported are:

- Enveloped
- Enveloping
- WSS 2004
- WSS 1.1
- Attachments
- Signed WSS SwA attachments
- ebXML signatures with SOAP attachments

## Key Types and Profiles Supported

The key types supported are:

- RSA, DSA, ECC
- PKCS#1, PKIPath, PKCS#7, X.509 BST Token Profile 1.1

## Signature Task Screen Terms

While signing a document, please consider the following terms and definitions:

TERM	DESCRIPTION
WSS 1.1	When selected, specifies the WSS 1.1 WS-Security specification for this signature to adhere to.
WSS 2004	When selected, specifies the WSS 2004 WS-Security specification for this signature to adhere to.
Enveloped Signature	When selected, adds Enveloped Signature. <b>Enveloped Signatures</b> are those signatures that are contained within the element being signed. In other words, the element includes the signature as content.

---

Enveloping Signature	When selected, adds Enveloping Signature. <b><i>Enveloping Signatures</i></b> are those signatures that wrap the document content, including any enveloped signatures, inside the enveloping signature element.
Transform	Default is Canonical method of transformation. Other methods of canonical transformations for signatures include Canonical XML with Comments, Exclusive Canonical XML and Exclusive Canonical XML with Comments.

---

TERM	DESCRIPTION
Use Key from Identified User	When selected, uses the signing key specified in the local system user policy for the user identified at run-time.
Use Static Key from Policy	When selected, applies Signature Policy highlighted in the Policy table for signing.
Signature policy	Name of the XML Signature Policy to apply for signing. Example: SIG_Danielle.
Sign Attachments	When checked, any SOAP attachments present in the message will be signed by the product according to Web Services Security SOAP Messages with Attachments (SwA) Profile 1.1 specification.
Filter embedded content signatures (not recommended)	<p>Check the Filter embedded content signatures (not recommended) checkbox only when it is known that at a later time another user will be inserting an additional enveloped signature within the content signed by this signature.</p> <p>When unchecked, any existing signatures in the content will be included in the current signature. This option should not be checked unless it is known that an additional enveloped signature will later be added within the current signed content. This option should never be checked for WSS signatures.</p>
Key Identifier	<ul style="list-style-type: none"> <li>• With None selected, the X.509 certificate used for signing is neither referenced nor included in the document. The recipient must use other means to obtain knowledge of the X.509 certificate.</li> <li>• With X.509 selected, the complete X.509 is used for identifying the key.</li> <li>• With SerialNumber selected, the X.509 issuer DN and serial number is used for identifying the key.</li> <li>• With SubjectKeyIdentifier selected, the X.509 v3 SubjectKeyIdentifier extension is used for identifying the key.</li> </ul>
Elements to Sign Path	Node/sub-node selected for signing.

### Canonicalizing XML Signatures

Canonicalization normalizes XML documents by removing possible variations in the document such as insignificant white space and new lines so that any inconsequential changes made while processing the document do not impact the verification of the document.

The defaults depend on the sample document. Try a soap document to see the more common defaults.



The default settings for canonicalization are required in the XML Signature Specification, and also support interoperability with external systems that support signature verification.

Defaults in the SIGN screen are:

- Type – WSS 1.1
- Transform – Canonical XML

Under the Transform drop down listing, options available for subsequent signatures being applied within the signed data of prior signatures are:

OPTION LABEL	W3C DSIG SPECIFICATION REQUIREMENT	XML COMMENTS	CONTEXT-SENSITIVITY	RESTRICTIONS ON SIGNATURES
Canonical XML	Yes	No	Yes	Not within or above *
Canonical XML with Comments	No	Yes	Yes	Not within or above *
Exclusive Canonical XML	No	No	No	Not within signed data
Exclusive Canonical XML with Comments	No	Yes	No	Not within signed data

“Not within or above” means that a signature added later to an ancestor element of the content, i.e. an element that at some level contains the signed element, or the element or a descendant of the element, may invalidate the initial signature.

### Signature Transform Definitions

The **Canonical XML** option tells both the signer and the verifier to canonicalize the document and signature prior to signing and verification. The transform instructions are included in the signature. XML comments are not signed or verified. The signature is context-sensitive and the signed data cannot be wrapped after signing, e.g. in an additional enveloping signature or SOAP message.

The **Canonical XML with Comments** option is the same as Canonical XML, but includes XML comments in the signing and verification process. Normally XML comments are not signed or verified. The specification recommends that vendors support Canonical XML with comments as an option, but support for this option is not required.

The **Exclusive Canonical XML** option is a context-insensitive version of canonicalization. With ordinary canonicalization you cannot generally wrap the signed XML data with new XML tags. For example, if you add an enveloped canonical signature on an element, then add an enveloping signature to the document, the enveloped signature will not verify because the XML context of the signed element will have been changed by the wrapping of the entire document with the enveloping signature. XML context is part of an ordinary canonical signature. (Technically, it is the namespaces of higher-level elements that are included in the signature). Similarly, all ordinary canonical signatures would likely fail to verify if the document was wrapped in a SOAP message.

Exclusive canonical XML excludes the XML context from the signing and verification so that the signature will still verify even if the signed element is later signed with an enveloping signature, wrapped in a SOAP message, or otherwise modified with respect to context. Exclusive canonical XML allows changes to the document outside the signed data, but not inside the signed data. Exclusive canonical XML is a new specification and may not be supported by all vendors.

The **Exclusive Canonical XML with Comments** option is the same as Exclusive Canonical XML, but includes comments in the signing and verification process. Normally XML comments are not signed or verified. Exclusive canonical XML is a new specification and may not be supported by all vendors.

### Filter Embedded Content Signatures Checkbox Definitions

Options that apply to the Filter embedded content signatures (not recommended) checkbox are:

- **Unchecked** includes any other signatures present in the signed data in the signing and verification process. If additional signatures are added within the signed data, verification is not possible. For example, multiple enveloped signatures cannot be added to the same element.
- **Checked** excludes all signatures present in the signed data from the signing and verification process. This option allows multiple enveloped signatures to be applied to the same data. New signatures may be applied within the signed data of prior signatures.

**Note:** Forum Systems recommends that you use the following signature properties options:

- Transform – Canonical XML
- Sign Signatures – checked

### Apply ebXML Signatures with SOAP Attachments

When adding an enveloped signature to the SOAP Envelope or SOAP Header of an incoming request that includes an ebXML MessageHeader SOAP header, the product detects the ebXML and applies an ebXML-compliant signature to the document. These actions are performed by the product automatically during the Sign Document task at the content-level.

## TASK: VERIFY DOCUMENT SIGNATURE

The Verify Document Signature task may be used to verify digital signatures to ensure integrity and support authentication and non-repudiation. A digital signature may cover one, multiple, or all portions of an XML document and attachments. The specific portions of the document requiring signature may be specified. The Verify Document Signature task uses the private key or Signer Group specified in a Verification Policy to verify signatures and can enforce algorithm restrictions.

The specifications supported on the system for the Verify Document Signature task are:

- W3C XML-Signature Syntax and Processing
- W3C Canonical XML Version 1.0
- W3C Exclusive XML Canonicalization Version 1.0
- OASIS WS-Security 1.1
- OASIS WS-Security SOAP Messages with Attachments Profile 1.1
- OASIS WS-Security 2004
- OASIS ebXML Message Service 2.0

### Signature Types Supported for Verification

The signature types supported for the Verify Document Signature task are:

- XML digital signatures
- WS-Security signatures
- ebXML signatures
- WS-Security SOAP attachment signatures

### Verify ebXML Signatures

The system supports ebXML-compliant signature verification. The configuration and verification of ebXML signatures are performed with the same steps used for other digital signatures supported by the system.

---

**VERIFY SIGNATURE**

---

Task Type: Verify Document Signature

Task Name\*:

On Error: ☒ Log & Halt Processing ☐ Log & Continue

---

**VERIFICATION PROPERTIES**

---

Verification Policy:  [Edit](#)

☐ Allow XPath and XSLT transforms (not recommended)

☐ Require signature on all attachments

☐ Remove signature

☐ Save certificate thumbprint

---

When ebXML signatures are present in the sample document, the **Allow XPath and XSLT Transforms** option is selected by default.

## Verify Attachments

The system supports verification of signatures in the document that apply to attachments. The configuration and verification of attachment signatures are performed with the same steps used for other digital signatures supported by the system.

### Option Available For Removing a Signature

Using the **Remove Signature** checkbox in the Verify Document Signature Task will remove any verified signature, including XML and WS-Security, and any ID or wsu:Id attributes inserted into the document during the Sign Document task. If the resulting WS-Security header is empty, this task will strip out the WS-Security. This task will not remove any security tokens in the WS-Security SOAP header and will not remove the WS-Security header if it is not empty.

---

**VERIFY SIGNATURE**

---

Task Type: Verify Document Signature

Task Name\*:

On Error: ☒ Log & Halt Processing ☐ Log & Continue

---

**VERIFICATION PROPERTIES**

---

Verification Policy:  [Edit](#)

☐ Allow XPath and XSLT transforms (not recommended)

☐ Require signature on all attachments

☒ Remove signature

☐ Save certificate thumbprint

---

**Note:** To remove both the WS-Security X.509s and signature-related ID attributes, use both of the following settings:

- the Remove Signature checkbox in the Verify Document Signature task
- the Remove WS-Security Header task.

### Verify Signature with Allow XPath and XSLT Transforms Option

The Allow XPath and XSLT transform option allows the signer to use xpath and xslt transforms to exclude content within signed elements from the signature. Excluding content from the signature may allow tampering and repudiation. Certain specifications, e.g. ebXML, require the use of xpath or xslt transforms. When allowing xpath and xslt transforms in signatures, additional measures should be used to verify that the xpath and xslt transform expressions used are consistent with established security policies. The Identify Document task can be used, for example, as a primitive check that xpath and xslt expressions are provided exactly as expected.

### Option Available Requiring Signatures on All Attachments

Using the **Require signature on all attachments** checkbox in the Verify Document Signature Task will verify that all attachments included in an XML or WSDL policy are signed.

## Verify Document Signature Task Screen Terms

While verifying the signature on a document, please consider the following terms and definitions:

TERM	DESCRIPTION OF OPTIONS
<b>VERIFY SIGNATURE</b>	
On Error	<ul style="list-style-type: none"><li>• With Log &amp; Halt Processing selected, if an error is encountered, the verification process will log an error and halt processing.</li><li>• With Log &amp; Continue selected, if an error is encountered, the verification process will log an error and continue processing.</li></ul>
<b>VERIFICATION PROPERTIES</b>	
Verification policy	Specifies the Verification Policies to use.
Allow XPath and XSLT transforms (not recommended)	When checked, the system allows XPath and XSLT transforms to exclude content within signed elements from the signature before verification occurs.
Require signature on all attachments	When checked, verifies that all document attachments are signed.
Remove signature	When checked, removes both the WS-Security signatures and signature-related ID attributes.
<b>REQUIRED SIGNATURES</b>	
Path	Node selected for verification options.
<b>ELEMENTS REQUIRING SIGNATURE</b>	
Path	Node that the verified signature must reference and include in the signed content

## TASK: VIRUS SCAN

The virus scan task enables scanning for malware threat vectors within any target content including headers, body, attachments, and embedded malware within JSON or XML structured elements or attributes.

TASK LISTS > TASK LIST: NEW TASK LIST2 > TASK: VIRUS SCAN

**VIRUS SCAN**

Task Type:Virus Scan

Task Name\*:Virus Scan

☒ Scan Incoming Document

☒ Scan attachments

☐ Add Virus Scanned Header

Action when a virus is found: ☒ Block transaction

☐ Replace content with

☐ Remove content

☐ Flag transaction

Encoding of elements: ☒ Base64 ☐ Hex Binary

**SELECT ELEMENTS TO DECODE**

☐ soap:Envelope

- ☐ soap:Body

Elements to Decode and Virus Scan

☐ ELEMENT

No items to display

Apply Save

### Virus Scan Task Screen Terms

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
Scan Incoming Document	Include the inbound document body as a target for the virus scan engine
Scan Attachments	Include attachments as documents to scan for malware. This option will automatically detect the file type and for files such as ZIP files will open and scan for malware within the ZIP archive.

Elements to Base64 Decode and Virus Scan	Used to target selective sections of the XML/SOAP document where BASE64 data will be present and has the potential to be malware.
---	--

## TASK: ZIP CONTENTS PROCESSING

This task requires that you have the ZIP feature enabled in your license. The task allows for ZIP file contents to be processed through the Task List processing.

TASK LISTS > TASK LIST: PROCESS ZIP ATTACHMENT > TASK: ZIP CONTENTS PROCESSING

---

**ZIP CONTENTS PROCESSING**

Task Type: ZIP Contents Processing

Task Name\*: ZIP Contents Processing

On Error: ☒ Log & Halt Processing ☐ Log & Continue

Task List Group\*: tlg\_Process\_ZIP\_Contents [Edit](#)

☐ Replace document with Task List output

☐ Encryption Policy: Encryption\_Policy\_Attribute (RSA, AES-256) [Edit](#)

☐ Decryption Policy: Decryption\_Policy\_service [Edit](#)

☐ Debug Logging:

Allow up to 10 zip entries

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

To enable ZIP processing, you will create a Task List Group with a Task List that contains the ZIP Contents Processing task. The ZIP Contents Processing task in turn will require a Task List Group that has a task list to be used to process each of the ZIP file entries.

### Virus Scan Task Screen Terms

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
On Error	Determine action to take if an error occurs in processing
Task List Group	The Task List Group to be used to process the ZIP file entries
Replace document with Task List output	When checked this option will replace the contents of the ZIP file entry with the output of the associated task list
Encryption Policy	When checked this option will use the encryption policy to encrypt the ZIP file entry
Decryption Policy	When checked this option will use the decryption policy to decrypt the ZIP file entry
Debug Logging	When checked this option will increase the level of logging for the ZIP file processing to verbose
Allow up to [ ] zip entries	Used to limit the number of ZIP entries allowed to be processed



## ZIP Content Processing Request Filter Requirement

In order to use the ZIP Content Processing task the request filter policy associated with the WSDL or Content Policy that handles the request or response document needs to be set up with the format 'Multipart' setting such as what is shown in the screenshot below.

REQUEST FILTER POLICIES > REQUEST FILTER POLICY > MESSAGE TYPE FILTER

**MESSAGE TYPE FILTER**

**Name\*:**

MTOM

**Format:**

Multipart

**Description:**

SOAP Message Transmission Optimization Mechanism

**Identification Expression\*:**

(Content-Type ==~ "(?i)multipart/related.\*type=application/xop+xml.\*") && (method == "POST")

☐ Generate Expression

**Methods:**

☐ GET

☐ POST

☐ HEAD

☐ PUT

☐ DELETE

☐ OPTIONS

☐ TRACE

☐ CONNECT

**Content Types:**

☐ ANY

☐ XML

☐ SOAP 1.1

☐ SOAP 1.2

☐ SwA

☐ MIME

☐ MTOM

☐ DIME

☐ JSON

☐ URL Encoded

☐ Web Form

**Parameter:**

**Map to Attributes:**

☐

**Remote Convert Content-Encoding:**

[No conversion]

**Client Convert Content-Encoding:**

[No conversion]

Apply

Save

## APPENDIX

### Appendix A - Constraints in Tasks Management Guide

ELEMENT	CONSTRAINTS	CHARACTER COUNT
Task name	Case sensitive, alphanumeric characters, may be from 1-256, and allows dashes, hyphens and spaces.	1-256
Task List name	Unique and case sensitive. Allows dashes, hyphens and spaces.	1-256
Task List Group name	Unique and case sensitive. Allows dashes, hyphens and spaces.	1-256
Constant field in the SAML Attribute Value Type dialog	Allows any keyboard character.	1-256
Attribute name used for Attribute replacement variables	Unique and case insensitive. May include the following characters:  A-Z a-z 0-9 ! # \$ % & ' * + - . ^ _ `   ~	Unlimited

## Appendix B - Encrypt Screen Reference Chart in Tasks Management Guide

When applying an encryption, the ENCRYPT screen presents the options below:

The screenshot shows the ENCRYPT screen with the following sections and options:

- Task Type:** Encrypt Elements
- Task Name:** Encrypt Elements
- On Error:** ☒ Log & Halt Processing ☐ Log & Continue
- ENCRYPTION PROPERTIES**
  - Type:** ☒ Encrypt Element ☐ Encrypt Content
  - Method:** ☐ WSS 1.1 ☐ WSS 2004 ☒ XML Encryption
  - Encryption policy:** ENC\_Danielle (AES-192)
  - ☐ Encrypt attachments
  - Key Identifier:** ☒ SerialNumber ☐ X.509 ☐ SubjectKeyIdentifier ☐ Subject
- SELECT ELEMENTS TO ENCRYPT**
  - ☒ Invoice
    - ☐ InvoiceNo
    - ☐ OrderDate
    - ☒ Item
      - ☒ ProductID
      - ☒ Price
      - ☐ Quantity
- Elements to Encrypt**
  - ☐ PATH
  - ☐ Invoice/ItemPrice
  - ☐ Invoice/ItemProductID

Buttons at the bottom: Remove, Apply, Save.

**Annotations (Left Side):**

- Select an Error Handling option; Log & Halt Processing or Log & Continue.
- Select Encrypt Element or Encrypt Content as the type of encryption to apply.
- Select WSS 1.1, WSS 2004 or XML Encryption for this encryption to adhere to.
- Select the XML Encryption Policy to apply to the selected nodes.
- Use Encrypt Attachments checkbox to add the selected encryption to any attachments present in the message.
- Key identification provides options for:
  1. Select SerialNumber which uses the X.509 issuer DN and serial number.
  2. Select X.509 which uses the complete X.509.
  3. Select SubjectKeyIdentifier which uses the X.509 v3 SubjectKeyIdentifier extension.
  4. Select Subject which uses the X.509 subject DN.
- Check the Select All checkbox ( ☒ ) to select all the elements for encryption.
- The Path field displays the nodes/subnodes that are selected for encryption.

Figure 1: Options Available in the Encrypt Screen.

## Appendix C - Signature Screen Reference Chart in Tasks Management Guide

When applying a signature, the SIGN screen presents many options visible below:

**SIGN**

Task Type: Sign Document  
Task Name\*: Sign Document

On Error: ☒ Log & Halt Processing ☐ Log & Continue

**SIGNATURE PROPERTIES**

Type: ☒ WSS 1.1 ☐ WSS 2004 ☐ Enveloped Signature ☐ Enveloping Signature

Transform: Canonical XML

☐ Use key from identified user  
☒ Use static key from policy  
Signature policy: SIG\_Jack (RSA)

☐ Sign attachments  
☐ Filter embedded content signatures (not recommended)

Key Identifier: ☐ None ☒ X.509 ☐ SerialNumber ☐ SubjectKeyIdentifier

**SELECT ELEMENTS TO SIGN**

☐ soap:Envelope  
☒ soap:Body

**Elements to Sign**

☐ PATH  
☐ /soap:Envelope/soap:Body

Remove Apply Save

**Annotations:**

- Either select the Log & Halt or Log & Continue option for error handling on this task.
- Select WSS 1.1, WSS 2004, Enveloped Signature or Enveloping Signature for the signature type.
- Select a Signature Transformation option from the Transform drop down list.
- Select Use Key from Identified User option to apply key from identified user revealed in the User ID & Access Control task.
- Select Use Static Key from policy option to apply key selected in Signature Policy drop down list (SIG\_Danielle) for signing.
- Select the Sign Attachments checkbox to include a signature of the attachments present in the message in the selected signature.
- Select the Filter embedded content signatures (not recommended) checkbox only when it is known that at a later time another user will be inserting an additional enveloped signature within the content signed by this signature.
- When unchecked, any existing signatures in the content will be included in the current signature. This option should not be checked unless it is known that an additional enveloped signature will later be added within the current signed content. This option should never be checked for WSS signatures.
- Key identification provides options for:
  1. Select None which uses no key identifier.
  2. Select X.509 which uses the complete X.509.
  3. Select SerialNumber which uses the serial number of the X.509.
  4. Select SubjectKeyIdentifier which uses the X.509 v3 SubjectKeyIdentifier extension.
- The Path field displays the node/subnode that is selected for signing (/soap:Envelope).
- The selected XML Signature Policy (SIG\_Jack) will be used to sign the selected node (/soap:Envelope/soap:Body).

Figure 2: Options Available in the Signature Screen.

**Note:** When signing a Document with attachments, the signatures of the attachments are also inserted into the document. The attachments themselves are not modified.

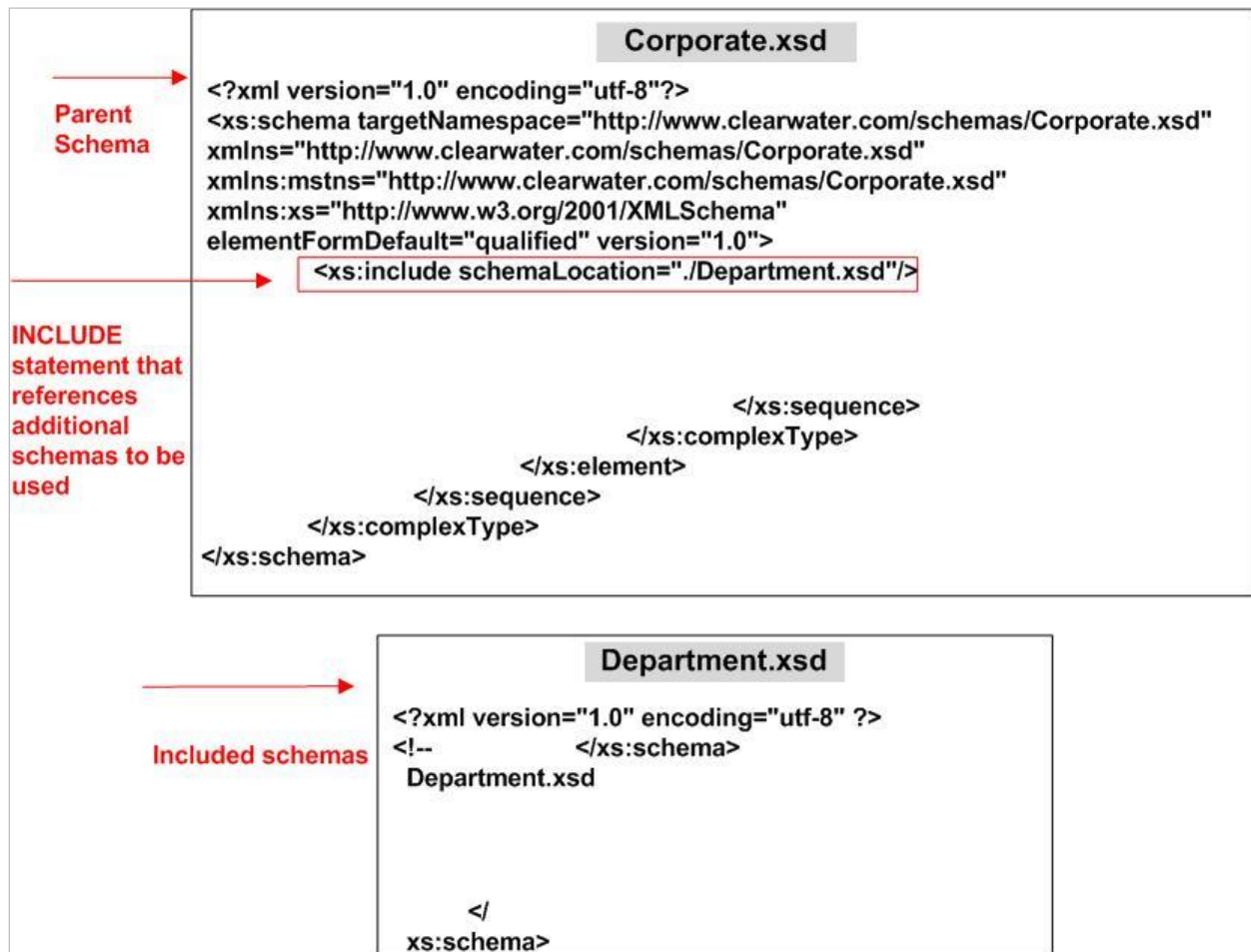
## Appendix D - Example Compound Schema Reference Chart in Tasks Management Guide

In this example, the parent schema is named `Corporate.xsd` and the child schema is named `Department.xsd`. Although this example is shown with only one included schema, a parent schema may have one or more included schemas.

The parent schema (`Corporate.xsd`) references an additional schema with an include statement, such as:

```
<xs:include schemaLocation="./Department.xsd"/>
```

as the following graphic displays:



**Figure 3: Example Compound Schema.**

A valid document is a document conforming to the defined DTD or XSD schema schema.

## INDEX

- .dtd schemas, 47
- .xsd schemas, 47
- .xsl or .xslt style sheets, 25
- Abort Processing task
  - in WSDL policy, 40
- Access Control List, 36
- ACL, 36
- Action, 60
- add a WS-Security Header with a Username Token
  - cautionary, 62, 63
- add replay verification with WS-Security Header Username Token
  - cautionary, 62, 63
- add User Identity/Access Control by Digital Signature, 39
- add User Identity/Access Control by protocol, 38
- add User Identity/Access Control by XML Mapping, 39
- Archive Document task, 53
- Assertion expires and Time, 58
- back end web server not WS-Security-aware
  - removing WS-Security Header, 23
- Canonical XML option, 76
- Canonical XML with Comments option, 76
- compound schema
  - example of, 86
- configuration options for SAML Assertion, 57
- configuration options for User Identity and Access Control, 35
- configuration options for WS-Security Headers, 60
- constant
  - Map Attributes to XML task, 30
- Constant, 59
- Constant field, 59
- conventions used, 1
- Cookie, 59
- Credential binding, 36
- CRL Distribution Points
  - Map Attributes to XML task, 30
- Decrypt screen terms, 21, 53, 55, 64
- Disallow caching of this assertion, 58
- DN, 59
- Documents
  - loading a sample document from an XML file, 2
- Documents screen
  - setting sample document as system default sample document, 4, 51
- Dynamic, based on established identity, 59
- Dynamic, based on protocol certificate, 59
- ebXML-compliant signature, 77
- Email, 59
- Encrypt Attachments
  - used with encryptions, 13, 69
- Encrypt Elements task
  - key identifiers for, 68
- Encrypt screen terms, 12, 41, 42, 43, 69
- Enveloped Signature
  - used with signatures, 74
- Enveloped Signatures, 74
- Enveloping Signature
  - used with signatures, 74
- Enveloping Signatures, 74
- example of compound schema, 86
- Exclusive Canonical XML with Comments option, 77
- Filter embedded content signatures option with signatures, 75
- Generate SAML Assertion
  - Signature Policy, 59
  - Statement Types, 59
- Generate SAML Assertion option
  - Assertion expires, 58
  - Disallow caching of assertion, 58
  - Email Identification Format, 58
  - Include identifier format URI, 59
  - Include validity start time, 58
  - Issuer, 58
  - X.509 DN, 58
- Include a validity start time and Time to start, 58
- Include certificate, 59
- include statements and Validate Document Structure task, 47
- Include the client IP address, 59
- Include the identifier format URI, 59
- Issuer, 58
- key identifiers in Encrypt Elements task, 68
- lax schemas, 47
- load a sample document from an XML file, 2
- Map Attributes to XML task
  - constant, 30
  - CRL Distribution Points, 30
  - Netscape Certificate Comment, 30
  - Netscape Certificate Type, 30
  - protocol header, 30
  - user attributes, 30
  - X.509 attributes, 30
- Namespace, 59, 60
- Netscape Certificate Comment
  - Map Attributes to XML task, 30
- Netscape Certificate Type
  - Map Attributes to XML task, 30

- nonce
  - validity range for, 62, 63
- options for transforming Signatures, 76
- Override remote routing
  - Remote Routing task, 44
- Path
  - used with decryptions, 65
  - used with elements requiring verification, 80
  - used with encryptions, 69
  - used with signatures, 75
  - used with verifications, 80
- Pattern Match Control Flow, 72
- Pattern Match screen terms, 70, 71
- prerequisites for User Identity/Access Control, 37
- prerequisites for WS-Security Headers, 62
- Process Attachment Task terms, 22
- protocol header
  - Map Attributes to XML task, 30
- Query Data Source Task terms, 31, 32, 33
- Remote Routing task
  - Override remote routing, 44
  - Replace message with remote response, 44
  - Send asynchronous message copy, 44
- remove WS-Security Header
  - when back end web server not WS-Security-aware, 23
- removes signature
  - in WS-Security Header task, 23
- Replace message with remote response
  - Remote Routing task, 44
- Resource, 60
- restrictions on signatures, 76
- Run Task List, 5
- SAML 1.1 specification, 58
- SAML 2.0 specification, 58
- SAML Assertion
  - configuration options for, 57
- SAML Attribute
  - Namespace, 59
  - Value
    - Constant field, 59
    - Constant option, 59
    - Value type, 59
- SAML Attribute Value
  - Cookie option, 59
  - DN option, 59
  - Email option, 59
  - User attribute option, 59
  - User name option, 59
- SAML Authentication
  - include client IP address, 59
- SAML Authorization
  - Action, 60
  - Namespace, 60
  - Resource, 60

- SAML Email Identification option
  - dynamic, based on established identity, 59
  - static, based on specified user, 59
- SAML Signature Property option
  - Include certificate, 59
- schema
  - .xsd and .dtd, 47, 48
  - compound, strick or lax, 47
  - content-level decryption, 64
  - standalone, strick or lax, 47
  - with content-level encryption, 66
  - with element-level decryption, 64
  - with element-level encryption, 66
- Send asynchronous message copy
  - Remote Routing task, 44
- Serial Number
  - key identifiers in Encrypt Elements task, 68
- set sample document as system default sample document, 4, 51
- Sign Attachments
  - used with signatures, 75
- SIGN screen terms, 74
- Signature policy
  - used with signatures, 75
- Signature Policy, 59
- signatures added later to an ancestor element, 76
- Static, based on a specified user, 59
- strict schemas, 47
- style sheets
  - .xsl and .xslt, 25
- subject
  - key identifiers in Encrypt Elements task, 68
- SubjectKeyIdentifier
  - key identifiers in Encrypt Elements task, 68
- Task List, 5
  - use Run Task List, 5
- Task List Group, 6
- Tasks
  - Archive Document, 53
  - Transform Document, 25
  - User Identify & Access Control, 36
- terms
  - in Decrypt screen, 21, 53, 55, 64
  - in Encrypt screen, 12, 41, 42, 43, 69
  - in Pattern Match Task screen, 70, 71
  - in Process Attachments Task screen, 22
  - in Query Datasource Task screen, 31, 32, 33
  - in SIGN screen, 74
  - in Verify Signature screen, 80
- Transform
  - used with signatures, 74
- Transform Document task, 25
- transform Signature options, 76
- Use Key from Identified User
  - used with signatures, 75

- Use Static Key from Policy
  - used with signatures, 75
- User attribute, 59
- user attributes
  - Map Attributes to XML task, 30
- User Identity and Access Control
  - configuration options for, 35
- User Identity and Access Control Task, 35
- User Identity/Access Control
  - adding protocol-based, 38
  - by Digital Signature, 39
  - by XML Mapping, 39
  - prerequisites for, 37
- User Identity/Access Control task, 36
- User name, 59
- Validate Document Structure task and include statements, 47
- Value Type, 59
- Verify Signature screen terms, 80
- W3C DSig Specification, 76
- WSDL policy
  - Abort Processing task, 40
- WSS 1.1
  - used with encryption policies, 69
  - used with signatures, 74
- WSS 2004
  - used with encryption policies, 69
  - used with signatures, 74
- WS-Security Header
  - adding with Username Token and replay verification, 62, 63
  - apply no token, 63
  - apply X.509 binary token, 63
  - prerequisites for, 62
- WS-Security Header mustUnderstand attribute
  - WS-Security-aware, 62
- WS-Security Header task
  - removing signature from, 23
- WS-Security Header with Username Token
  - adding replay verification, 62, 63
- WS-Security Headers
  - configuration options for, 60
- WS-Security-aware
  - with WS-Security Header mustUnderstand attribute, 62
- X.509
  - key identifiers in Encrypt Elements task, 68
- X.509 attributes
  - Map Attributes to XML task, 30
- X.509 Distinguished Name, 58
- X.509 Identification option
  - dynamic, based on protocol certificate, 59