Forum Systems Sentry™ Version 8.1

Tasks Management Guide

Legal Marks

Forum Systems Sentry™ Version 8.1 Tasks Management Guide, published June 2013.

D-ASF-SE-01816

Table of Contents

List of Figures

INTRODUCTION TO THE TASK MANAGEMENT GUIDE

Audience for the Task Management Guide

The *Forum Systems Sentry™ Version 8.1 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: **********

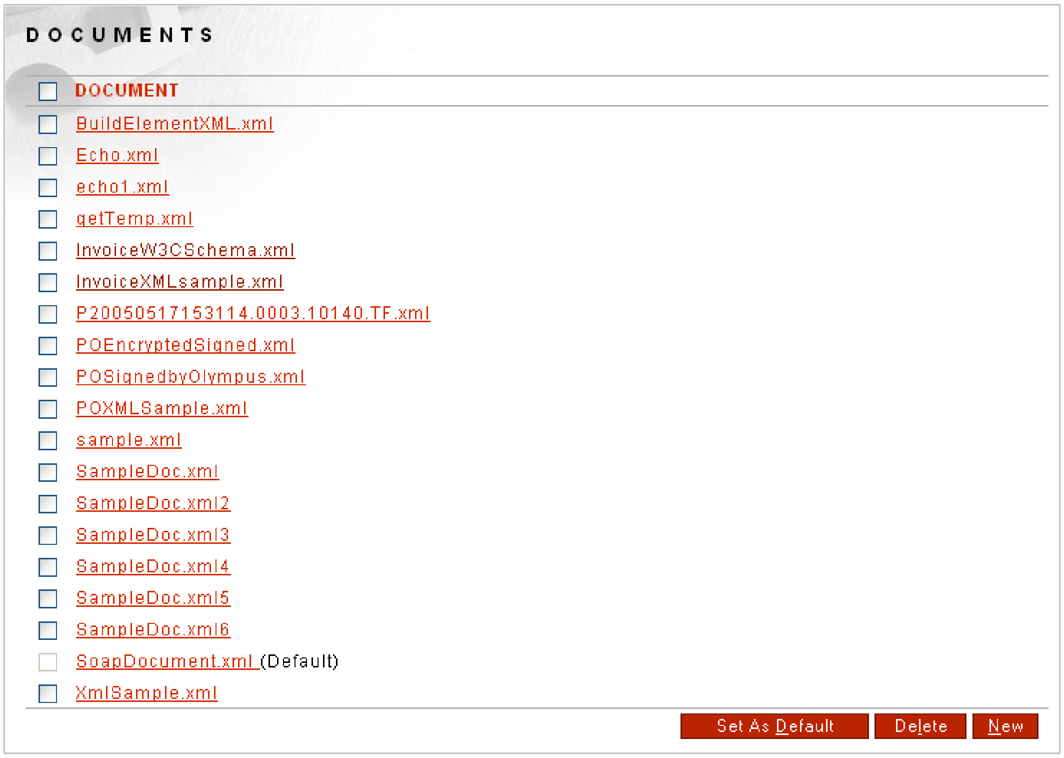
Customers with plug-in licenses on the system, confirm that your Integration License is visible on the General Info screen under the SUPPORTED FEATURES section.

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 XML files currently in the system.

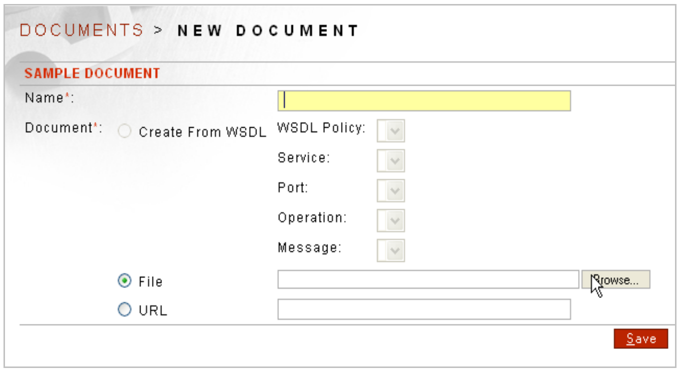


DOCUMENTS

- □ DOCUMENT
- □ BuildElementXML.xml
- □ Echo.xml
- □ echo1.xml
- □ getTemp.xml
- □ InvoiceW3CSchema.xml
- □ InvoiceXMLsample.xml
- □ P20050517153114.0003.10140.TF.xml
- □ POEncryptedSigned.xml
- □ POSignedbyOlympus.xml
- □ POXMLSample.xml
- □ sample.xml
- □ SampleDoc.xml
- □ SampleDoc.xml2
- □ SampleDoc.xml3
- □ SampleDoc.xml4
- □ SampleDoc.xml5
- □ SampleDoc.xml6
- □ SoapDocument.xml (Default)
- □ XmlSample.xml

Load a Sample Document from an XML File

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



DOCUMENTS > NEW DOCUMENT

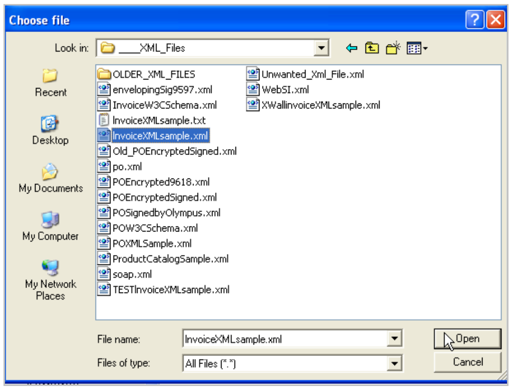
SAMPLE DOCUMENT

Name*:

Document*: ○ Create From WSDL WSDL Policy: Service: Port: Operation: Message:

● File

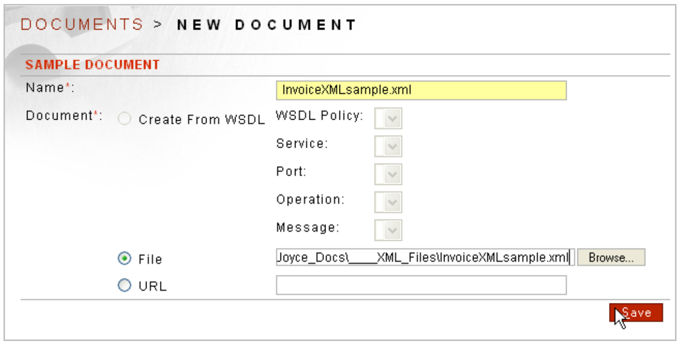
○ URL



Look in: ____XML_Files

File name: InvoiceXMLsample.xml

Files of type: All Files (*.*)



DOCUMENTS > NEW DOCUMENT

SAMPLE DOCUMENT

Name*: InvoiceXMLsample.xml

Document*: ○ Create From WSDL

● File Joyce_Docs____XML_Files\InvoiceXMLsample.xml

○ URL

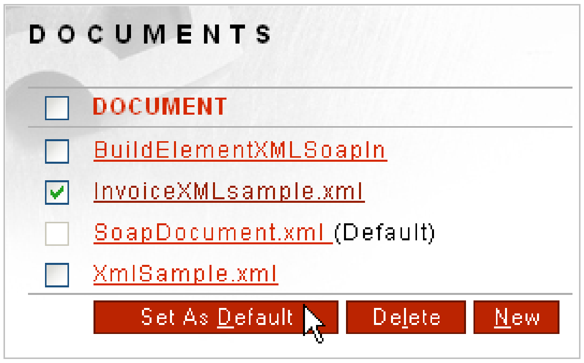
1. From the **GATEWAY** section of the Navigator, select **Documents** and the DOCUMENTS screen appears.
2. Select **New**, and the NEW DOCUMENT screen appears.
3. Select the **File** radio button, and then click **Browse**. The Choose file screen appears.
4. Navigate to and highlight an **XML file**. 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 **GATEWAY** -> **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:



☐ DOCUMENT
☐ BuildElementXMLSoapIn
☑ InvoiceXMLsample.xml
☐ SoapDocument.xml (Default)
☐ XmlSample.xml

- From the **GATEWAY** 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 contain one or more task list. Task list groups are associated with transaction policies such as HTML, XML, or WSDL policies in order to execute task lists within the task list groups.

Task Lists contain tasks which are specific actions that are performed on the request or response. The tasks available within Sentry include:

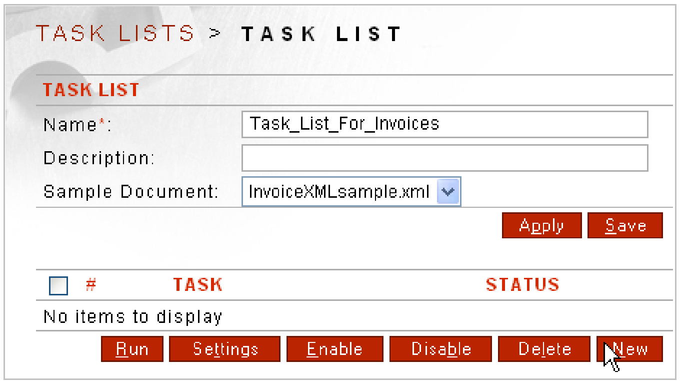
- Abort Processing
- Archive Document
- Convert JSON
- Convert SOAP to XML
- Convert XML to SOAP
- Convert XML Node
- Delay Processing
- Decrypt Elements
- Display WSDLs URIs
- Encrypt Elements
- Enrich Message
- Identify Document
- Logout
- Log
- Map Attributes to XML
- Map Attributes from XML
- Map 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

Sequenced Tasks 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 up or down arrows next to the task name.

Create a new Task

This instruction displays adding the Abort Processing task to the Task List:



TASK LISTS > TASK LIST

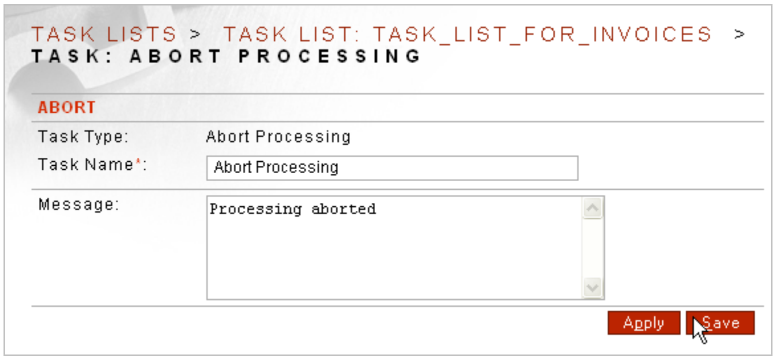
TASK LIST

Name*: Task_List_For_Invoices

Description:

Sample Document: InvoiceXMLsample.xml

□	#	TASK	STATUS
No items to display			



TASK LISTS > TASK LIST: TASK_LIST_FOR_INVOICES > TASK: ABORT PROCESSING

ABORT

Task Type: Abort Processing

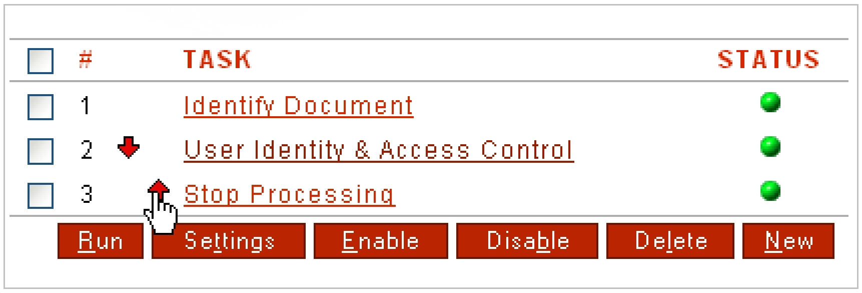
Task Name*: Abort Processing

Message: Processing aborted

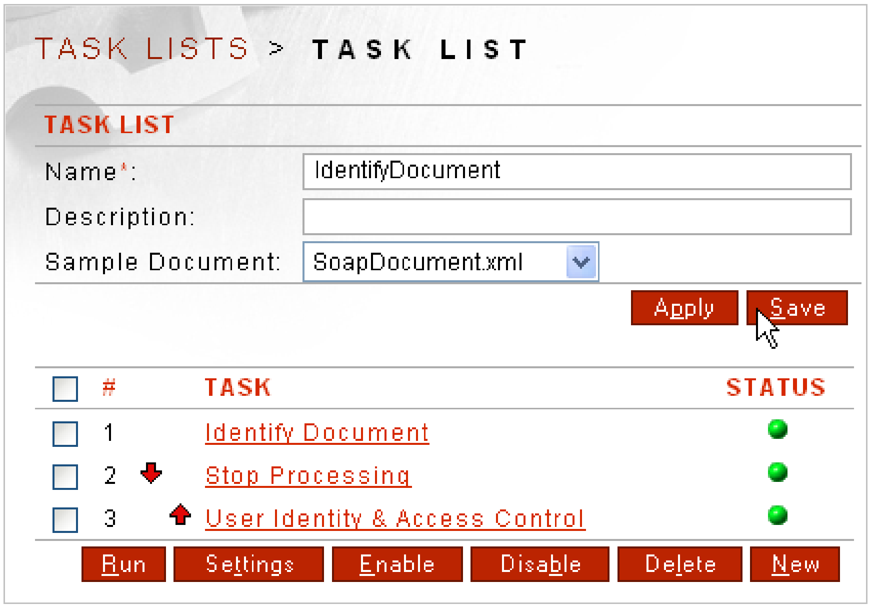
- From the **GATEWAY** section of the Navigator, select **Task Lists** and the TASK LIST screen appears.
- Select the **Task List name**, and the Task List appears.
- Retain the Sample Document (InvoiceXMLsample.xml) or select another **XML sample document name** from the Sample Document drop down list.
- Click **New** and the TASK TYPE screen appears. Select the **Abort Processing** radio button, and then click **Next**. The ABORT PROCESSING screen appears.
- Accept the pre-populated Task Name and Message, and then click **Save**. The TASK LIST screen refreshes with the “Configuration saved” message visible at the top of the screen.

Promote or Demote Tasks

Follow these steps to promote a Task in a given Task List:



□	#	TASK	STATUS
□	1	Identify Document	●
□	2	User Identity & Access Control	●
□	3	Stop Processing	●



TASK LISTS > TASK LIST

TASK LIST

Name*: IdentifyDocument

Description:

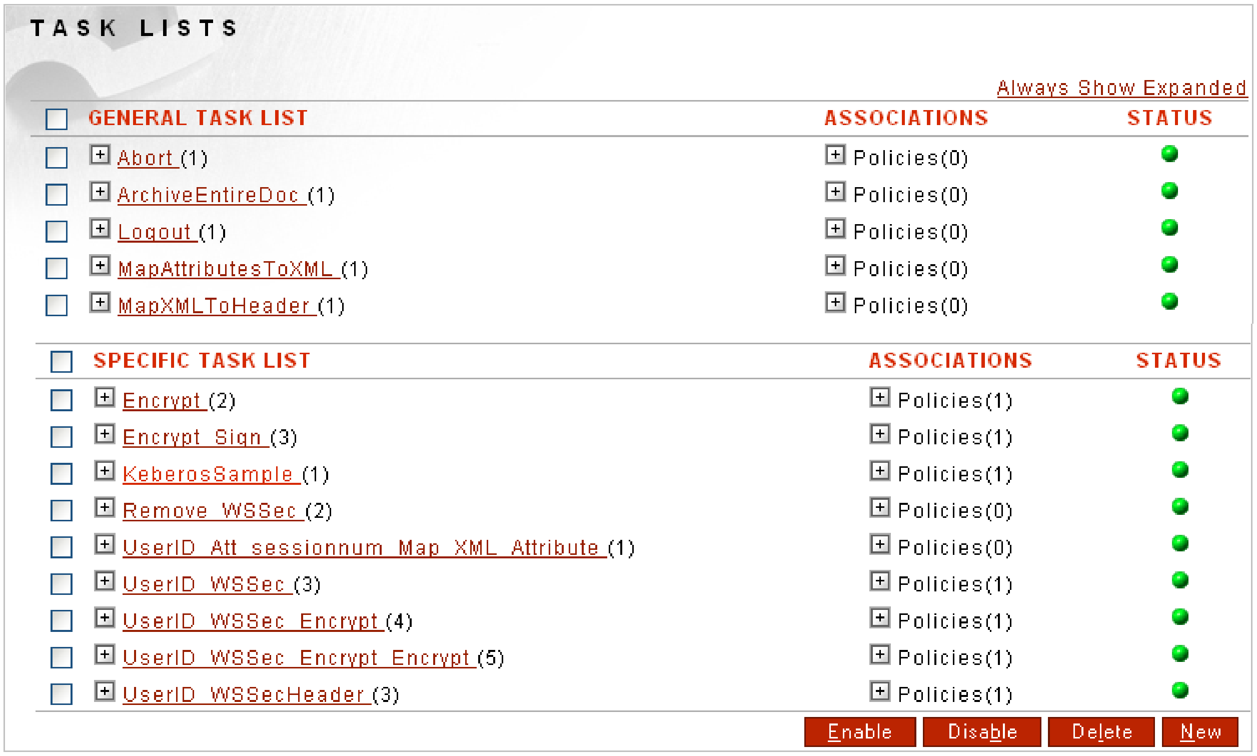
Sample Document: SoapDocument.xml

□	#	TASK	STATUS
□	1	Identify Document	●
□	2	Stop Processing	●
□	3	User Identity & Access Control	●

- From the **GATEWAY** section of the Navigator, select **Task Lists** and the TASK LIST screen appears.
- Select the **Task List name**, and the Task details screen appears.
- Click the **UP arrow** of a task to be promoted.
- On the refreshed TASK LIST screen, select **Save**.

TASK LISTS

A Task List is a grouping of sequentially ordered Tasks created on the WebAdmin UI. Task Lists are later consumed by XML or WSDL policies being processed by the system. Task Lists are reusable, so naming them with an easily-recognizable name is advisable.



□	GENERAL TASK LIST	ASSOCIATIONS	STATUS
□	⊞ Abort (1)	⊞ Policies(0)	●
□	⊞ ArchiveEntireDoc (1)	⊞ Policies(0)	●
□	⊞ Logout (1)	⊞ Policies(0)	●
□	⊞ MapAttributesToXML (1)	⊞ Policies(0)	●
□	⊞ MapXMLToHeader (1)	⊞ Policies(0)	●

□	SPECIFIC TASK LIST	ASSOCIATIONS	STATUS
□	⊞ Encrypt (2)	⊞ Policies(1)	●
□	⊞ Encrypt_Sign (3)	⊞ Policies(1)	●
□	⊞ KeberosSample (1)	⊞ Policies(1)	●
□	⊞ Remove_WSSec (2)	⊞ Policies(0)	●
□	⊞ UserID_Att_sessionnum_Map_XML_Attribute (1)	⊞ Policies(0)	●
□	⊞ UserID_WSSec (3)	⊞ Policies(1)	●
□	⊞ UserID_WSSec_Encrypt (4)	⊞ Policies(1)	●
□	⊞ UserID_WSSec_Encrypt_Encrypt (5)	⊞ Policies(1)	●
□	⊞ UserID_WSSecHeader (3)	⊞ Policies(1)	●

The Task Lists screen includes two major sections:

- General Task Lists - are those Task Lists that do not include the Identify Document task.
- Specific Task Lists - are those Task Lists that do include the Identify Document task.

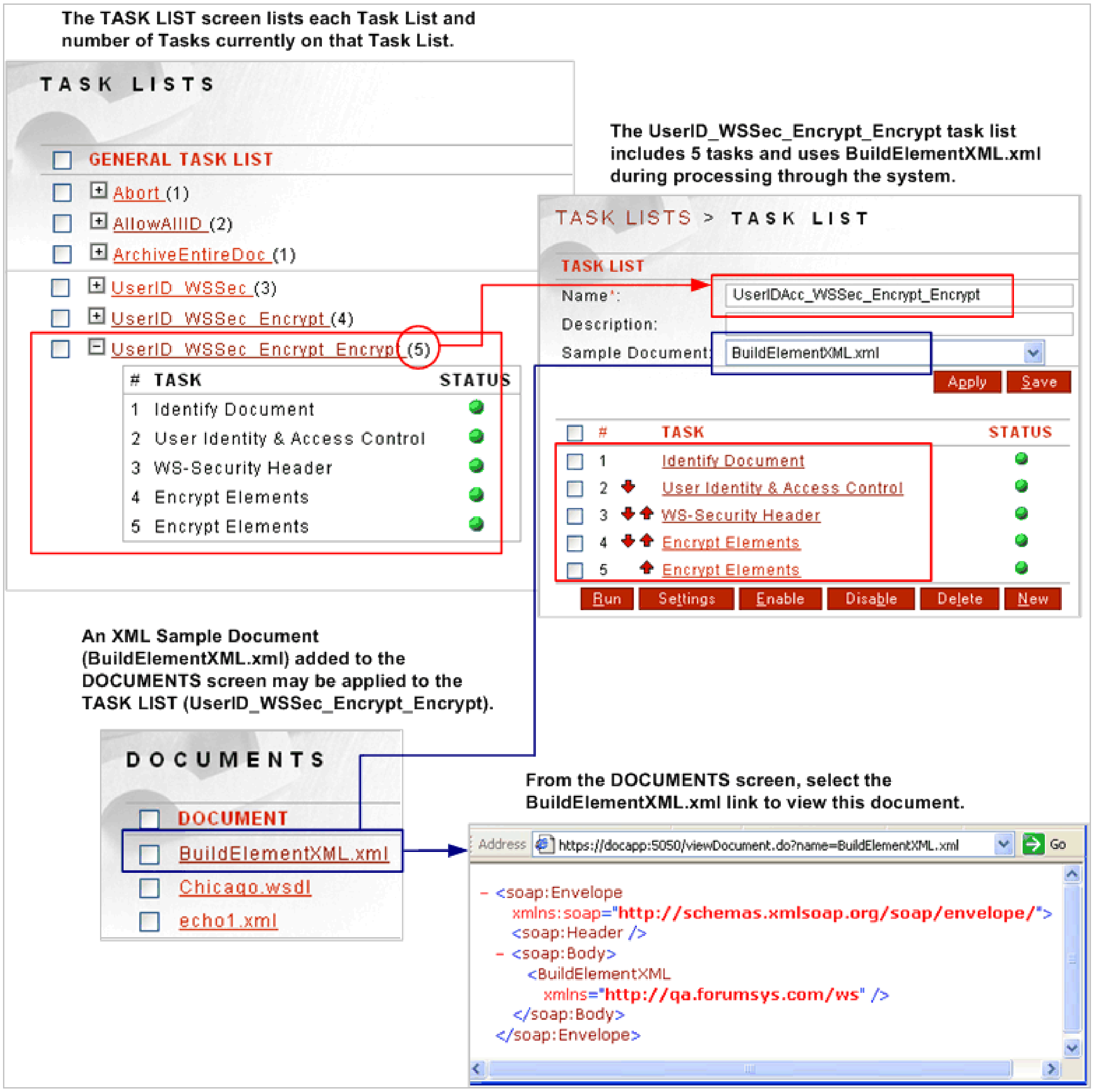
The Associations column tracks the number of other policies in the system that are referenced by this Task List.

Task List Examples

Examples for the Task List task include:

- Add a Task List.
- Run the Task List.
- Set Design-time Task Validation.

The following graphic displays the relationship between XML documents, Tasks and Task Lists.



The TASK LIST screen lists each Task List and number of Tasks currently on that Task List.

TASK LISTS

□ GENERAL TASK LIST

□ ⊞ Abort (1)

□ ⊞ AllowAllID (2)

□ ⊞ ArchiveEntireDoc (1)

□ ⊞ UserID_WSSec (3)

□ ⊞ UserID_WSSec_Encrypt (4)

□ ⊟ UserID_WSSec_Encrypt_Encryp (5)

#	TASK	STATUS
1	Identify Document	●
2	User Identity & Access Control	●
3	WS-Security Header	●
4	Encrypt Elements	●
5	Encrypt Elements	●

The UserID_WSSec_Encrypt_Encrypt task list includes 5 tasks and uses BuildElementXML.xml during processing through the system.

TASK LISTS > TASK LIST

TASK LIST

Name*: UserIDAcc_WSSec_Encrypt_Encrypt

Description:

Sample Document: BuildElementXML.xml

□	#	TASK	STATUS
□	1	Identify Document	●
□	2 ⬇	User Identity & Access Control	●
□	3 ⬇⬆	WS-Security Header	●
□	4 ⬇⬆	Encrypt Elements	●
□	5 ⬆	Encrypt Elements	●

An XML Sample Document (BuildElementXML.xml) added to the DOCUMENTS screen may be applied to the TASK LIST (UserID_WSSec_Encrypt_Encrypt).

DOCUMENTS

□ DOCUMENT

□ BuildElementXML.xml

□ Chicago.wsdl

□ echo1.xml

From the DOCUMENTS screen, select the BuildElementXML.xml link to view this document.

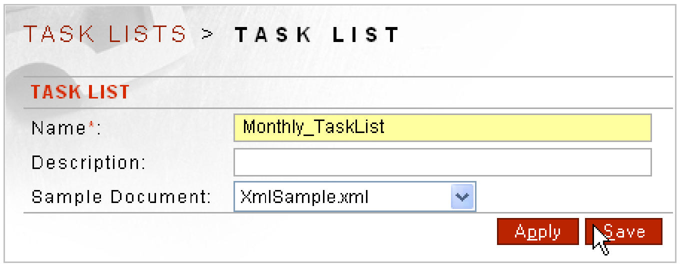
Address https://docapp:5050/viewDocument.do?name=BuildElementXML.xml

```
- <soap:Envelope
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Header />
  - <soap:Body>
      <BuildElementXML
        xmlns="http://qa.forumsys.com/ws" />
    </soap:Body>
  </soap:Envelope>
```

Figure 2: Relationship Among Task Lists, Tasks and XML Sample Documents.

Add a Task List

Follow these steps to add a Task List:



TASK LISTS > TASK LIST

TASK LIST

Name*: Monthly_TaskList

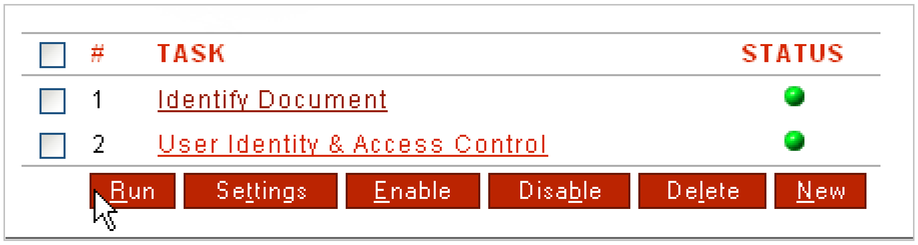
Description:

Sample Document: XmlSample.xml

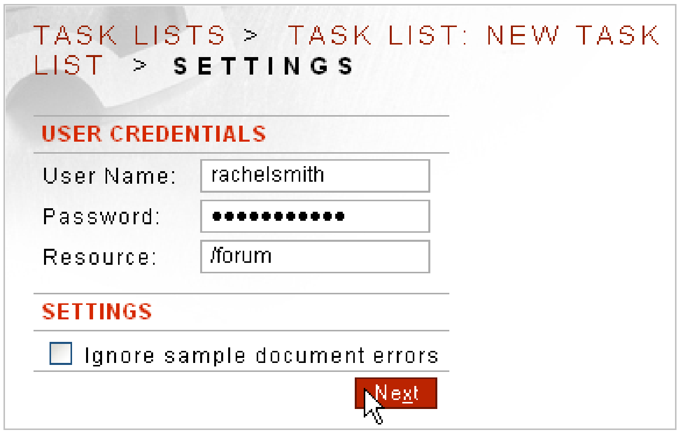
- From the **GATEWAY** section of the Navigator, select **Task Lists** and the TASK LIST screen appears.
- Overwrite the **value** in the Name field to a desired value.
- Enter a description in the Description field (optional).
- From the Sample Document drop down list, select a sample XML.
- Select **Save,** and the TASK LISTS screen appears with the “Task List added” message visible at the top of the screen.

Run the Task List

Administrators may use the “Run” command to verify settings and view the document after the system has processed the current Task(s). If a user must be identified, the USER CREDENTIALS screen appears.



□	#	TASK	STATUS
□	1	Identify Document	●
□	2	User Identity & Access Control	●



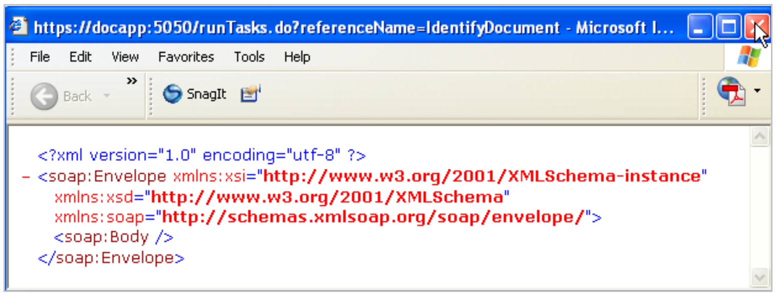
TASK LISTS > TASK LIST: NEW TASK LIST > SETTINGS

USER CREDENTIALS

User Name:	rachelsmith
Password:	●●●●●●●●●●●●
Resource:	/forum

SETTINGS

□ Ignore sample document errors



https://docapp:5050/runTasks.do?referenceName=IdentifyDocument - Microsoft I...
File
Edit
View
Favorites
Tools
Help
Back
SnagIt

```
  <?xml version="1.0" encoding="utf-8" ?>
- <soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body />
  </soap:Envelope>
```

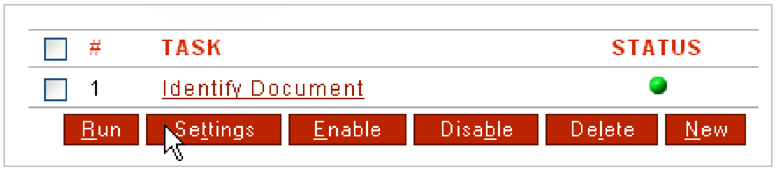
- Select **Run** from the TASKS screen.
- On the USER CREDENTIALS screen, in the User name field, enter a valid **User Name**.
- In the Password field, enter this valid User’s **Password.**

Note: The User Name and Password requested are from a User on the System; that is, a valid User who has membership in a valid Group that has membership in a valid ACL.

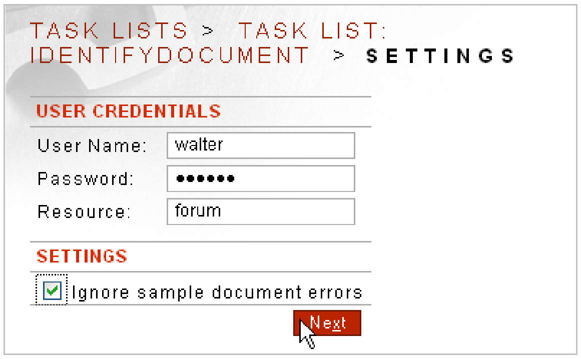
- In the Resource field, enter the **value** that represents a Resource in the system. The Resource field corresponds to the directory path where the Web Service will be deployed and may be used by external Identity Servers when performing authentication and authorization.
- Skip the Ignore a sample document errors checkbox, and then click **Next**.
- The read-only document appears. Click **OK** to close.

Set Design-time Task Validation

WebAdmin users may choose to ignore design-time errors processing sample documents when creating Task Lists. This feature may be used at design-time to ignore invalid signatures, expirations, and replays in sample documents and to specify XPaths that do not match the sample document. Follow these steps to apply a setting that allows the current WebAdmin user to ignore errors processing sample documents for all Task Lists:



□	#	TASK	STATUS
□	1	Identify Document	●



TASK LISTS > TASK LIST: IDENTIFYDOCUMENT > SETTINGS

USER CREDENTIALS

User Name: walter

Password: ••••••

Resource: forum

SETTINGS

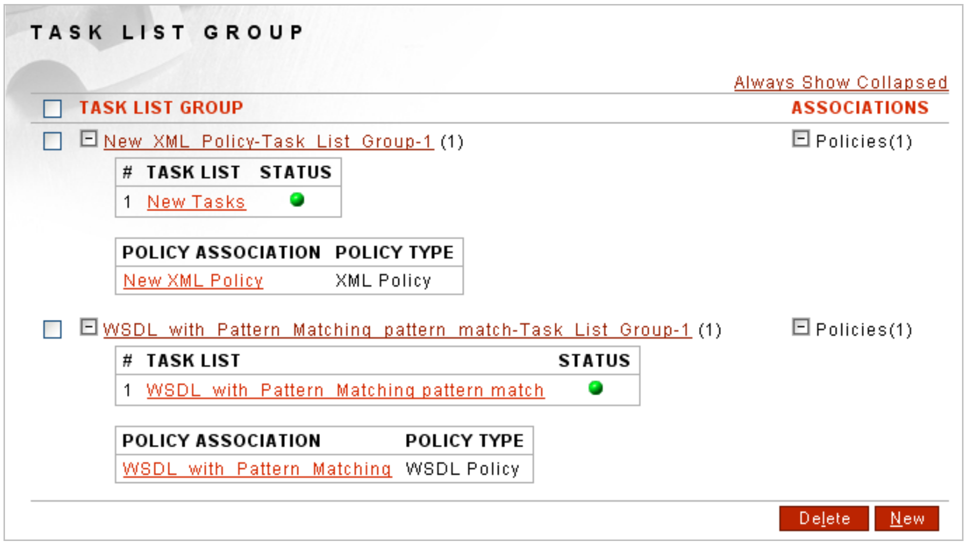
☑ Ignore sample document errors

- From the TASK screen, select **Settings**.
- Optionally specify a run-time **username**, **password**, and **resource** or skip those fields.
- Check the **Ignore a sample document errors** checkbox, and then click **Next**.
- The read-only document appears. Click **OK** to close.

Note: For testing purposes, you could generate a sample WSS Username token document with a nonce, for example, by running a WS-Security Username token task and saving the output. Subsequently, using the saved output as a sample document for a different Task List, you could check the **Ignore sample document errors** checkbox by selecting the **SETTINGS** button on the Task List screen. The new Task List could be run and configured without the hindrance of replay detection errors.

TASK LIST GROUPS

A Task List Group is a collective representation of one or more Task Lists created on the WebAdmin UI. The Task Lists in a Task List Group are later consumed by XML or WSDL policies being processed by the system. Task List Groups are reusable, so naming them with an easily-recognizable name is advisable.



TASK LIST GROUP

Always Show Collapsed

□ TASK LIST GROUP — ASSOCIATIONS

□ ⊟ New XML Policy-Task List Group-1 (1) — ⊟ Policies(1)

#	TASK LIST	STATUS
1	New Tasks	●

POLICY ASSOCIATION	POLICY TYPE
New XML Policy	XML Policy

□ ⊟ WSDL with Pattern Matching pattern match-Task List Group-1 (1) — ⊟ Policies(1)

#	TASK LIST	STATUS
1	WSDL with Pattern Matching pattern match	●

POLICY ASSOCIATION	POLICY TYPE
WSDL with Pattern Matching	WSDL Policy

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 WSDL or XML policy, apply the Task List Group.

Task List Group Examples

Examples for a Task List Group include:

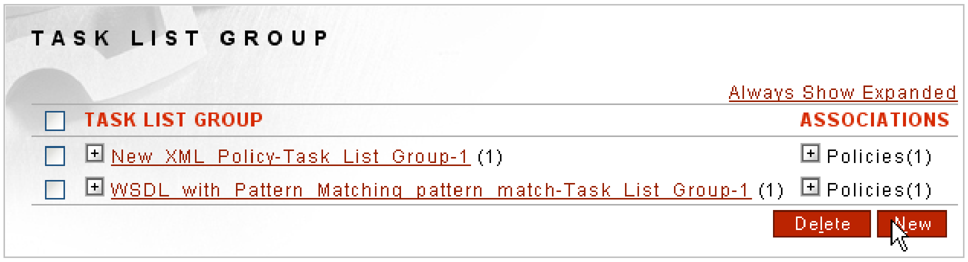
- Add a Task List Group
- Add a Task List to a Task List Group.

Note: For information on editing / viewing, deleting or removing a Task List Group, refer to the Common Operations section of the *Forum Systems Sentry™ Version 8.1 Web-based Administration Guide*. For information on promoting / demoting, or renaming a Task List Group, refer to the Tasks chapter of this document.

To associate a Task List Group to a WSDL or XML policy, refer to the *Forum Systems Sentry™ Version 8.1 WSDL Policies Guide* or *Forum Systems Sentry™ Version 8.1 XML Policies Guide* respectively.

Add a Task List Group

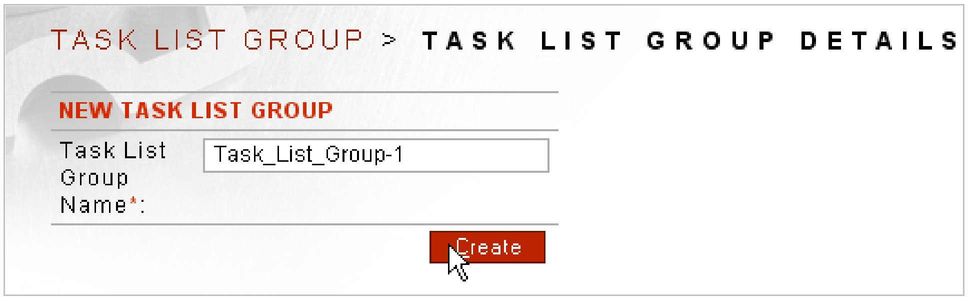
Follow these steps to add a Task List Group:



TASK LIST GROUP

Always Show Expanded

□ TASK LIST GROUP	ASSOCIATIONS
□ ⊞ New_XML_Policy-Task_List_Group-1 (1)	⊞ Policies(1)
□ ⊞ WSDL_with_Pattern_Matching_pattern_match-Task_List_Group-1 (1)	⊞ Policies(1)



TASK LIST GROUP > TASK LIST GROUP DETAILS

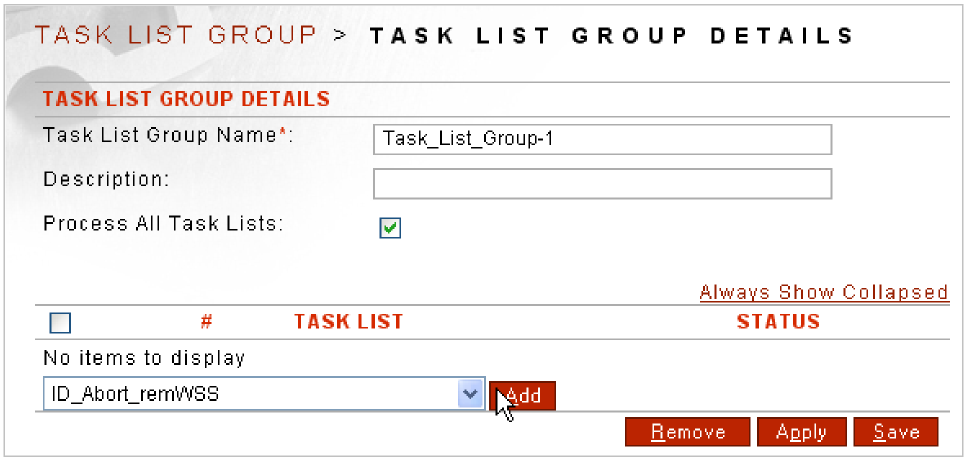
NEW TASK LIST GROUP

Task List Group Name*: Task_List_Group-1

- From the **GATEWAY** section of the Navigator, select **Task List Groups**.
- On the TASK LIST GROUP screen, in the Task List Group Name field, accept the name or enter a new **name**, and then click **Create**.

Add a Task List to a Task List Group

Users may add one or more Task Lists to a Task List Group. Follow these steps to add a Task List to the Task List Group:



TASK LIST GROUP > TASK LIST GROUP DETAILS

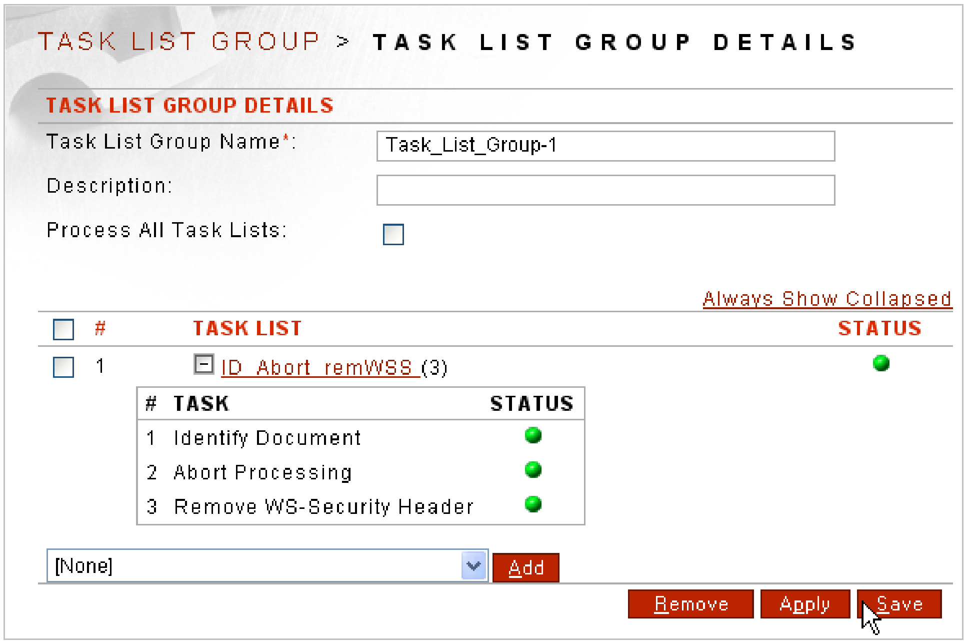
TASK LIST GROUP DETAILS

Task List Group Name*:	Task_List_Group-1
Description:	
Process All Task Lists:	☑

□	#	TASK LIST	STATUS

No items to display

ID_Abort_remWSS



TASK LIST GROUP > TASK LIST GROUP DETAILS

TASK LIST GROUP DETAILS

Task List Group Name*:	Task_List_Group-1
Description:	
Process All Task Lists:	□

□	#	TASK LIST	STATUS
□	1	⊟ ID_Abort_remWSS (3)	●

#	TASK	STATUS
1	Identify Document	●
2	Abort Processing	●
3	Remove WS-Security Header	●

[None]

- On the TASK LIST GROUP DETAILS screen, enter a **description** for this Task List Group in the Description field (optional).
- Check the **Process All Task Lists** checkbox to process all the Task Lists that are applied to this Task List Group through the system.
- From the TASK LIST drop down list, select a **Task List** to add to this Task List Group, and then select **Add**.
- On the refreshed screen, select **Save**.

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, HTML in order to accomplish the set of processing tasks for the transactions coming to that policy.

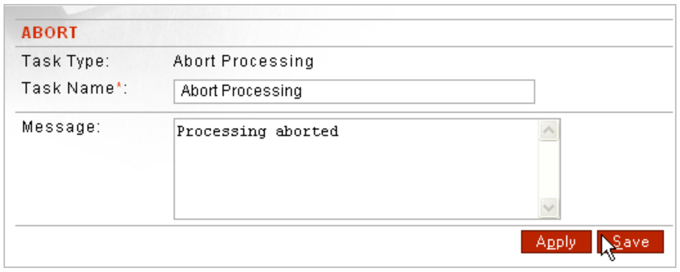
The tasks discussed in this section include:

- 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

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 with no coding. Document processing tasks provide comprehensive coverage across OASIS and W3C standards. Task processing provide integration capabilities with disparate vendors systems since Sentry can consume messages in any format, and convert them to 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. It is a core competency of the Sentry gateway.

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 Type: Abort Processing

Task Name*: Abort Processing

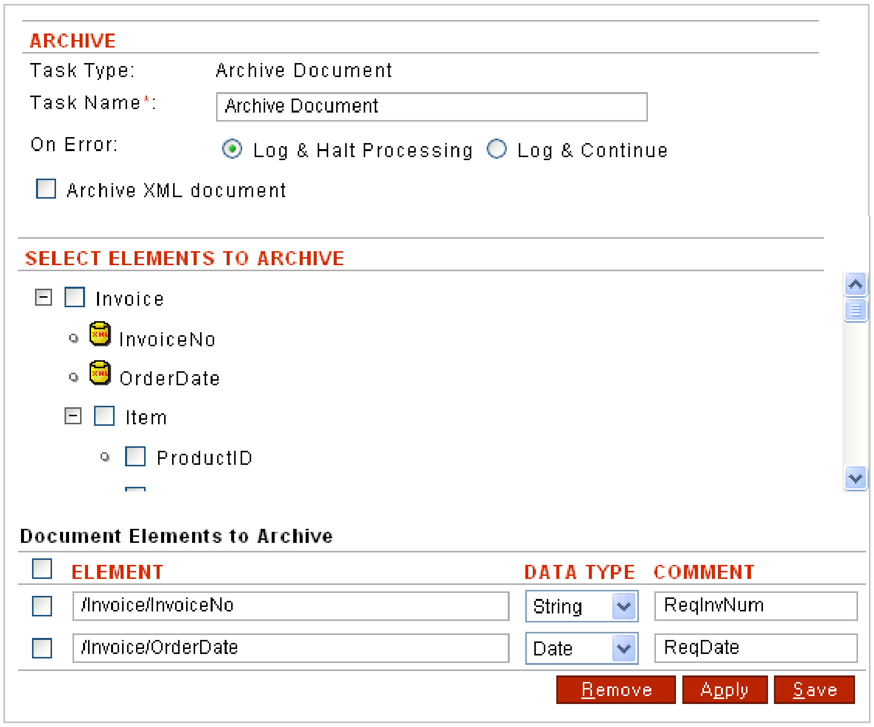
Message: Processing aborted

Abort Task Screen Terms

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

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

□	ELEMENT	DATA TYPE	COMMENT
□	/Invoice/InvoiceNo	String	ReqInvNum
□	/Invoice/OrderDate	Date	ReqDate

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

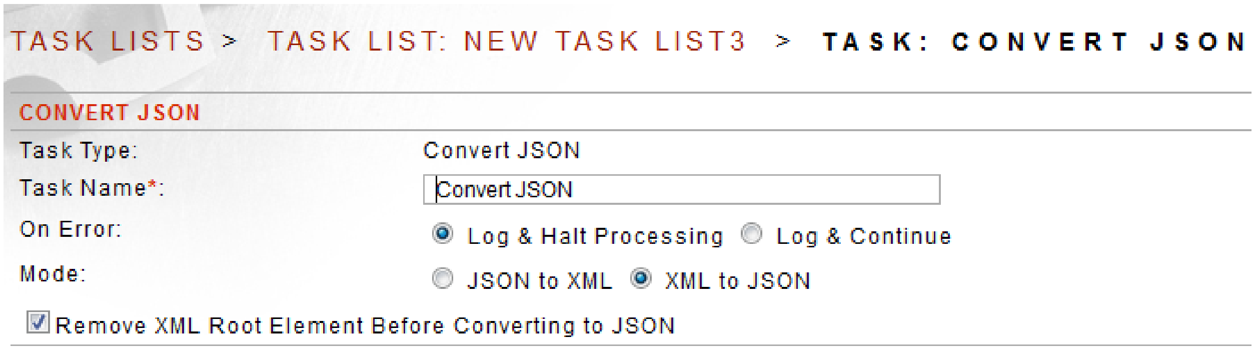
Archive Document Task Screen Terms

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: CONVERT JSON

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



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

CONVERT JSON

Task Type: Convert JSON

Task Name*: Convert JSON

On Error: ● Log & Halt Processing ○ Log & Continue

Mode: ○ JSON to XML ● XML to JSON

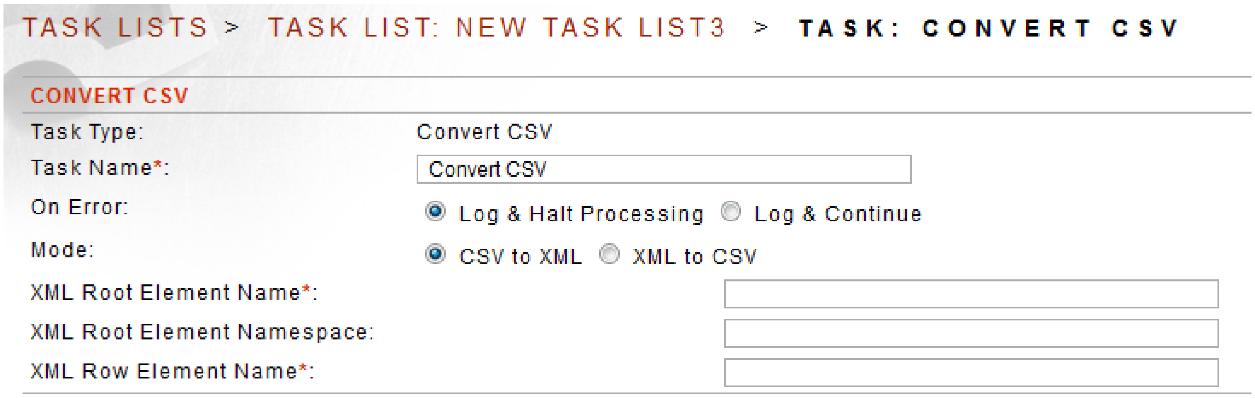
☑ Remove XML Root Element Before Converting to JSON

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 CSV

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



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

CONVERT CSV

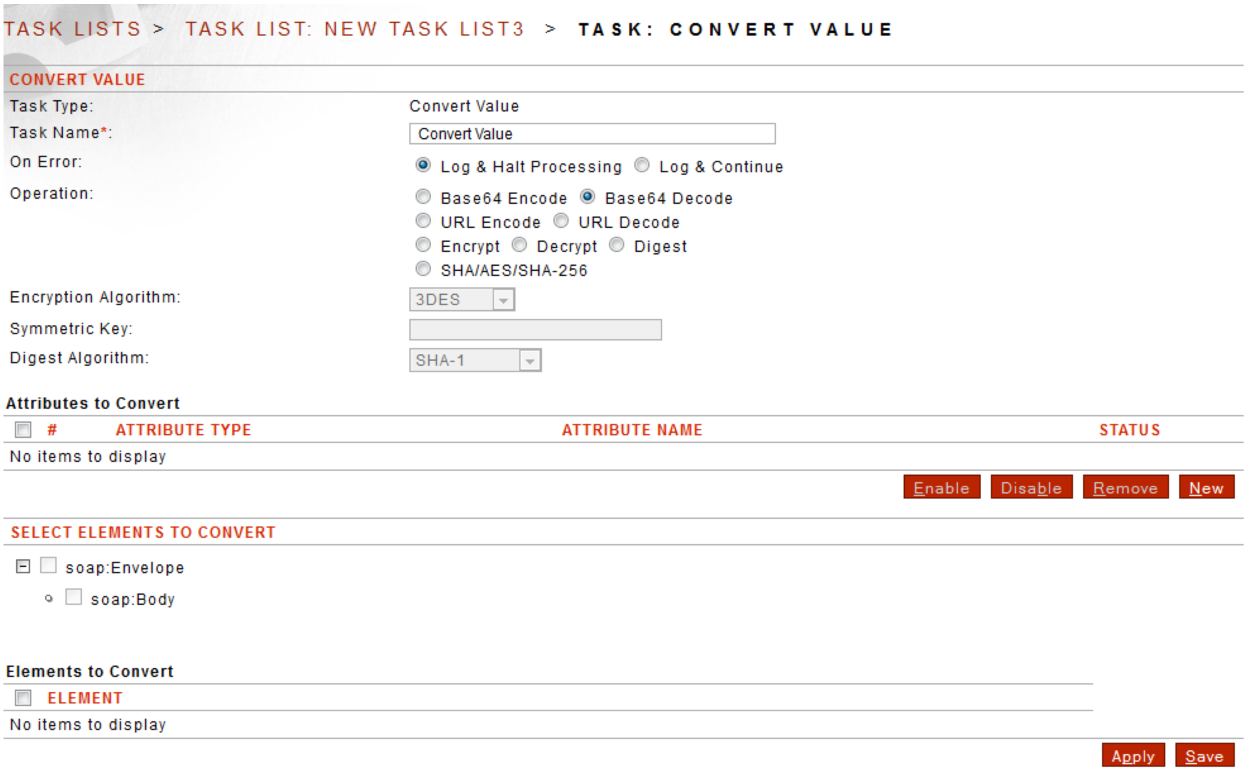
Task Type:	Convert CSV
Task Name*:	Convert CSV
On Error:	● Log & Halt Processing ○ Log & Continue
Mode:	● CSV to XML ○ XML to CSV
XML Root Element Name*:	
XML Root Element Namespace:	
XML Row Element Name*:	

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 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: NEW TASK LIST3 > TASK: CONVERT VALUE

CONVERT VALUE

Task Type: Convert Value

Task Name*: Convert Value

On Error: ● Log & Halt Processing ○ Log & Continue

Operation: ○ Base64 Encode ● Base64 Decode ○ URL Encode ○ URL Decode ○ Encrypt ○ Decrypt ○ Digest ○ SHA/AES/SHA-256

Encryption Algorithm: 3DES

Symmetric Key:

Digest Algorithm: SHA-1

Attributes to Convert

□ #	ATTRIBUTE TYPE	ATTRIBUTE NAME	STATUS
No items to display			

SELECT ELEMENTS TO CONVERT

□ soap:Envelope

□ soap:Body

Elements to Convert

□ ELEMENT
No items to display

Convert Value Task Screen Terms

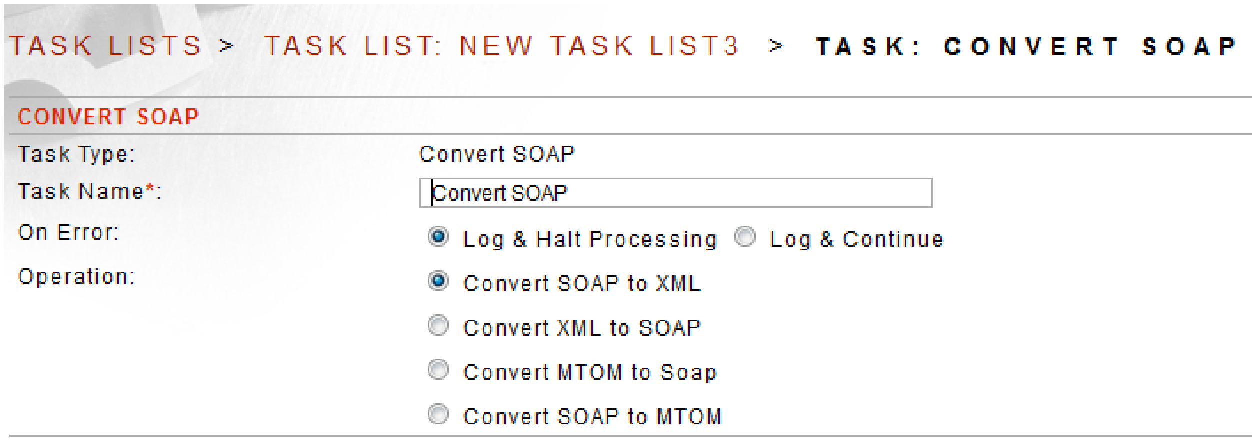
TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
On Error	Halt processing, or continue based on error in the task processing for this task
Operation	Base64 Encode Base64 Decode URL Encode URL Decode Encrypt Decrypt Digest SHA/AES/SHA-256
Encryption Algorithm	Enabled when Encrypt or Decrypt 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 Symettric 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 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
- **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: 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*:	Convert SOAP
On Error:	● Log & Halt Processing ○ Log & Continue
Operation:	● Convert SOAP to XML ○ Convert XML to SOAP ○ Convert MTOM to Soap ○ Convert SOAP to MTOM

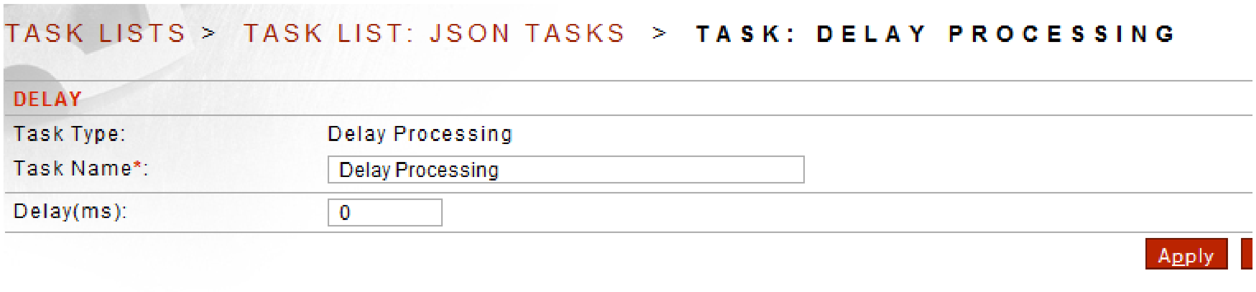
Convert SOAP To XML Task Screen Terms

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: 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*: Delay Processing

Delay(ms): 0

Delay Processing Screen Terms

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

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

TASK: DECRYPT ELEMENT

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.

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 furthr 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
DECRYPT	
On Error	• With Log & Halt Processing selected, if an error is encountered, the decryption process will log an error and halt processing. • With Log & Continue selected, if an error is encountered, the decryption process will log an error and continue processing.
DECYPTION PROPERTIES	
Decryption policy	A listing of current Decryption Policies to select from.
ELEMENTS TO DECRYPT	
Path	Node selected for decrypt options.

Decrypt Element Task Example

The example for the Decrypt Element tasks is Decrypt Elements.

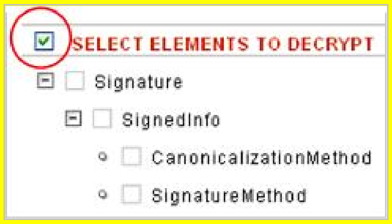
Decrypt an Element

A Key Pair policy is required for the Decrypt Task. When the product decrypts an XML Document, the system uses only the Private Key during this task. Follow these steps to decrypt an element in an XML policy:

Note: These operations assume that an Administrator has imported a PKCS Key Pair in the Keys screen, and then created an XML Decryption policy in the XML Decryption screen. For information on the Keys screen, refer to the *Forum Systems Sentry™ Version 8.1 Security Policies and PKI Guide*. For information on the XML Decryption screen, refer to the
XML Decryption Policies section of the *Forum Systems Sentry™ Version 8.1 XML Policies Guide*.

This operation also assumes that an Administrator has created an XML policy and created a Task List. With nested encrypted elements, you must decrypt the parent element first, then the next hierarchical lower element.

To select the entire document for decryption, check the Select All (□) prefacing SELECT ELEMENTS TO DECRYPT**. Example:**



☑ SELECT ELEMENTS TO DECRYPT

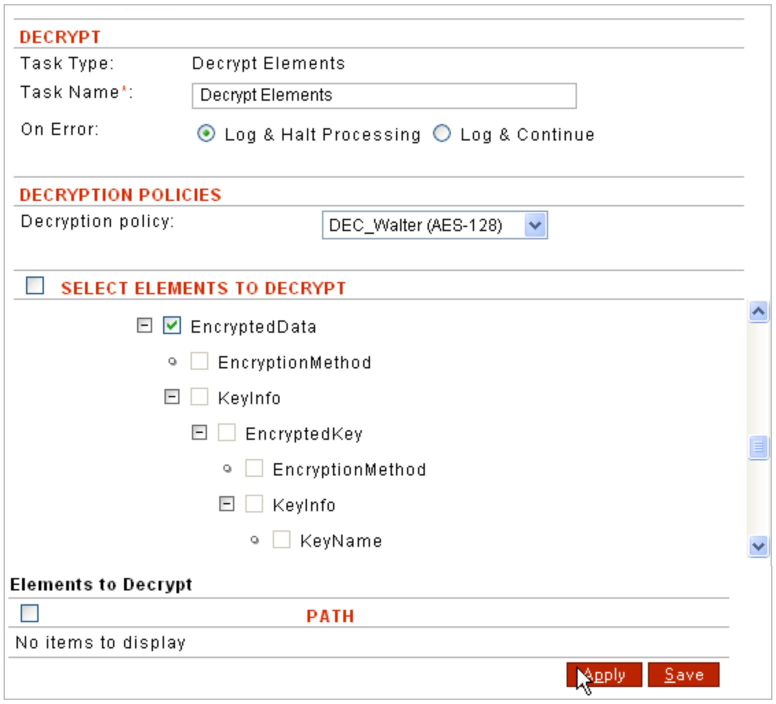
⊟ □ Signature

⊟ □ SignedInfo

○ □ CanonicalizationMethod

○ □ SignatureMethod

To select a single element for decryption, you would expand the elements under the SELECT ELEMENTS TO DECRYPT column, click on the target **element**, and then select **Apply**.



DECRYPT

Task Type: Decrypt Elements

Task Name*: Decrypt Elements

On Error: ● Log & Halt Processing ○ Log & Continue

DECRYPTION POLICIES

Decryption policy: DEC_Walter (AES-128)

□ SELECT ELEMENTS TO DECRYPT

⊟ ☑ EncryptedData

○ □ EncryptionMethod

⊟ □ KeyInfo

⊟ □ EncryptedKey

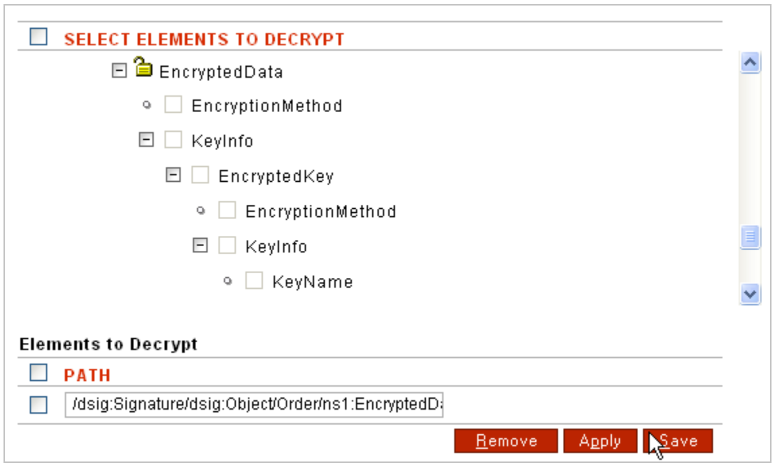
○ □ EncryptionMethod

⊟ □ KeyInfo

○ □ KeyName

Elements to Decrypt

□	PATH
No items to display	

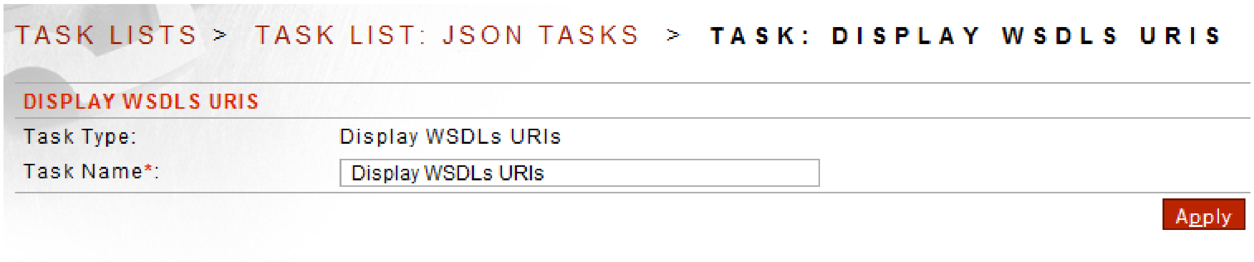


□ /dsig:Signature/dsig:Object/Order/ns1:EncryptedD

- From the TASK screen, select **New**.
- On the TASK TYPE screen, select the **Decrypt Element** link.
- On the DECRYPT screen, accept the pre-populated Task Name or overwrite the **name**.
- Aligned with On Error, select the **Log & Halt** Processing radio button.
- From the Decryption policy drop down list, select an **XML Decryption policy** to apply to this decryption.
- From the SELECT ELEMENTS TO DECRYPT section, expand elements and scroll down to view all possible encrypted elements.
- Check the **checkbox** prefacing the EncryptedData element, and then select **Apply**.
- The DECRYPT screen refreshes, the EncryptedData (the encrypted element) is now prefaced by the Open lock icon and the Elements to Decrypt area includes path for this element.
- Select **Save**.

TASK: DISPLAY WSDL URIs

This task is explicity 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*:	Display WSDLs URIs

Display WSDL URIs Task Screen Terms

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

TASK: ENCRYPT ELEMENTS

The following figure displays an Invoice issued from a Forum Systems owner to a customer, through the system:

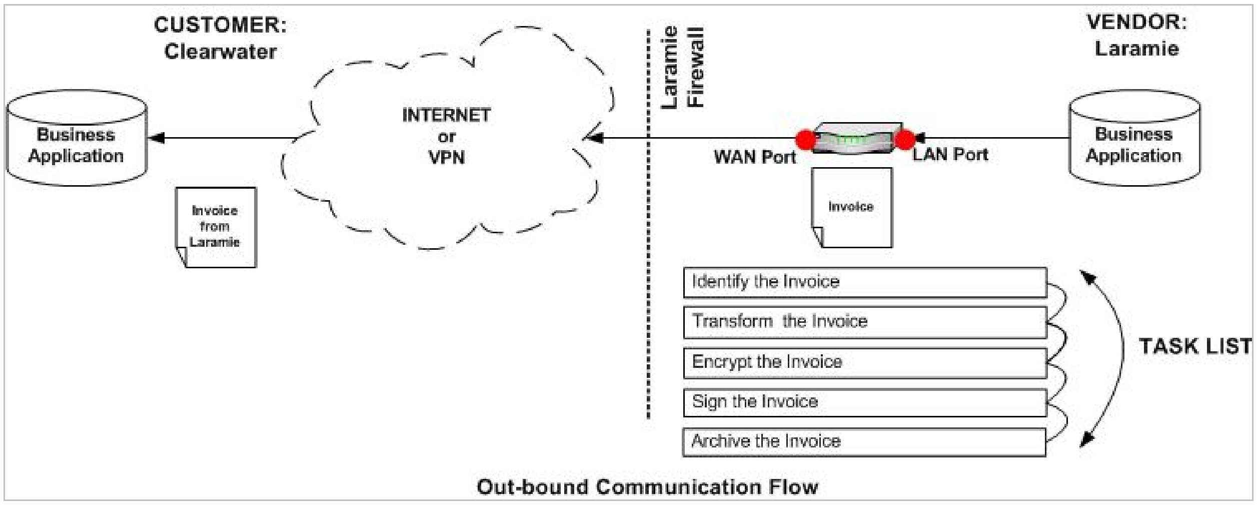


Figure 3: Invoice leaving the System.

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 encyption 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 8.1 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
ENCRYPT	
On Error	• With Log & Halt Processing selected, if an error is encountered, the encryption process will log an error and halt processing. • With Log & Continue selected, if an error is encountered, the encryption process will log an error and continue processing.
ENCRYPTION PROPERTIES	
Type	• With Encrypt Element selected, the encryption policy is applied to the entire node selected. • With Encrypt Content selected, the encryption policy is applied to the content of the node selected.
Method	• 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 • 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. • 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.
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	• With SerialNumber selected, the X.509 issuer DN and serial number is used for identifying the key. • With X.509 selected, the complete X.509 is used for identifying the key. • 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. • With Subject selected, the X.509 subject DN is used for identifying the key.
ELEMENTS TO ENCRYPT	
Path	Node selected for encrypt options.

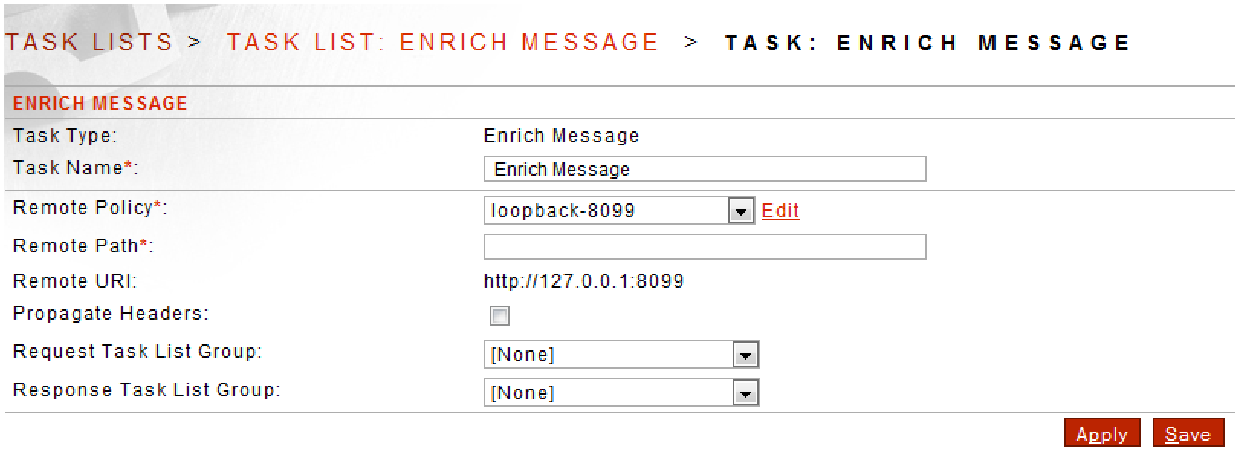
TASK: ENRICH MESSAGE

This task enables 3rd 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 seletion 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 3rd 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*:	Enrich Message
Remote Policy*:	loopback-8099 ▾ Edit
Remote Path*:	
Remote URI:	http://127.0.0.1:8099
Propagate Headers:	□
Request Task List Group:	[None] ▾
Response Task List Group:	[None] ▾

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 3rd 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: IDENTIFY DOCUMENT

The Identify Document task is used to designate which documents/transactions are to be identified as targetes 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.

Note: Tasks are not required to have an identify document rule defined. For tasks that do not have Identiy Document task defined will be applied to all documents sent to task list. If the Identify Document task is defined, the task will only trigger if the maching rules are the most specific rule set defined, and the rules match per the target transaction.

When using the Identify Document task, the top section of the screen (Header Filters) is for matching non-XML related items, the bottom section (Document Filters) is for matching XML content.

Identify Document Screen Terms

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

TERM	DESCRIPTION OF OPTIONS
IDENTIFY	
Task Name	Display name for the task
HEADER FILTERS	
Filter Type	The source of the input to target for the comparison. Sources include: • **Constant** o A static value specified directly on the policy • **Protocol Header** o 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** o The header from the inbound request from the client • **Response Header** o The header from the response from the back-end system • **User Attribute** o 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. • **X509 Attribute** o OID and other attributes within an X509 cetificate • **Query Parameter** o Each query string parameter from the inbound URI is available as an attribute to map • **Cookie** o A cookie from the header • **HTTP Method** o GET or POST • **Request URI** o The full URI of the inbound client request • **Username**

	o The currently authenticated client's username • **Source IP Address** o The current client's source IP • **HTTP Status Code** o The response code from the back-end system response back to Sentry
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	The target value for the identification comparison. Values Include: • **Protocol Header** o 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** o The header from the inbound request from the client • **Response Header** o The header from the response from the back-end system • **User Attribute** o A general attribute type the can be referenced by other tasks • **Query Parameter** o A target name value pair to add to the URI for the back-end server request from Sentry • **Cookie** o Using this setting will result in a SET-COOKIE header being created with the cookie value as the value specified. • **HTTP Method** o GET or POST • **Request URI** o The full URI of the inbound client request
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 1.1 format used to target the specified location within the XML document to identify based on content and the matching function.

TASK: LOGOUT

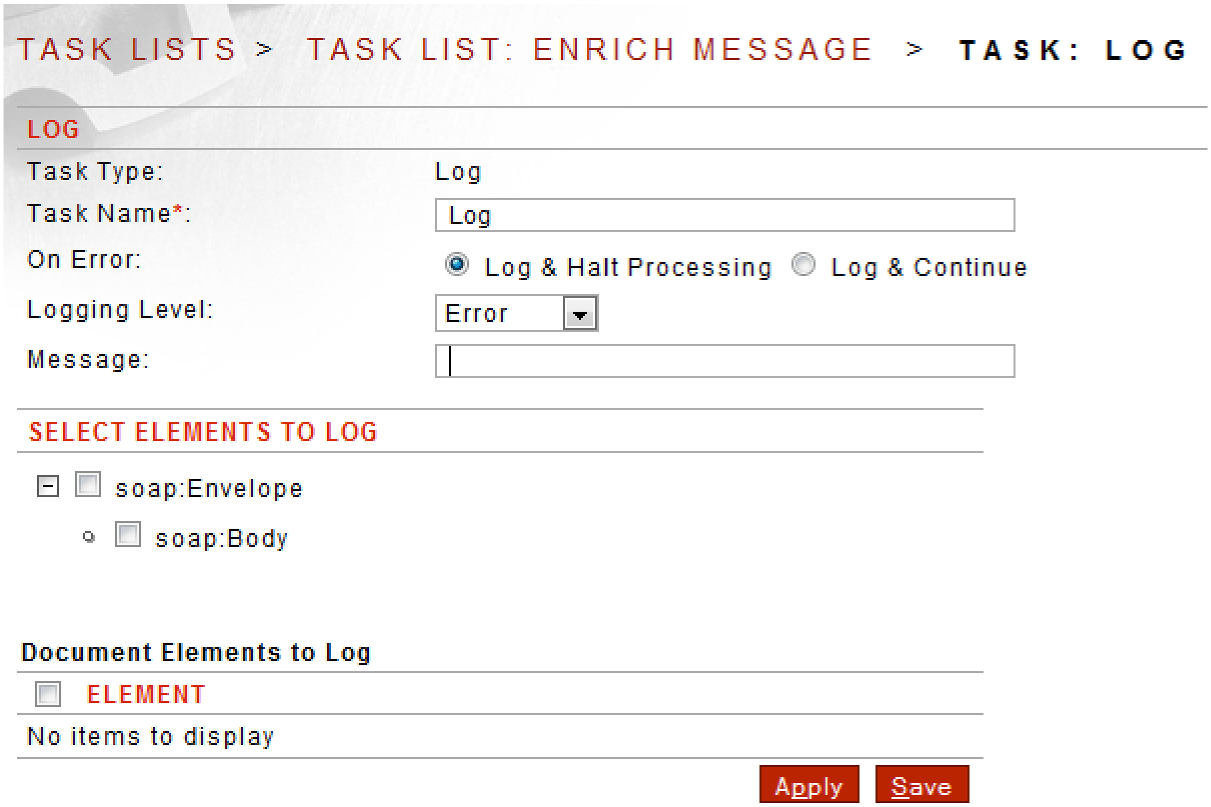
The Logout task is used for invalidating the session token for SiteMinder or the session Token for Forum Sentry or Forum STS tokens. This task requires persistent session caching is enabled on the SiteMinder policy server, or persistent sessions enabled on Forum Sentry or Forum STS Identity Broker. Upon 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: 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*: Log

On Error: ● Log & Halt Processing ○ Log & Continue

Logging Level: Error

Message:

SELECT ELEMENTS TO LOG

□ soap:Envelope

□ soap:Body

Document Elements to Log

□ ELEMENT

No items to display

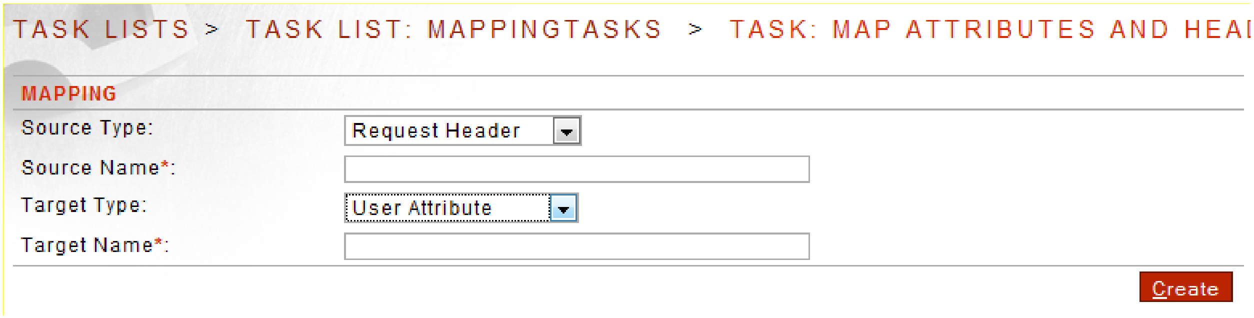
Log Message Screen Terms

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: 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 HEAI

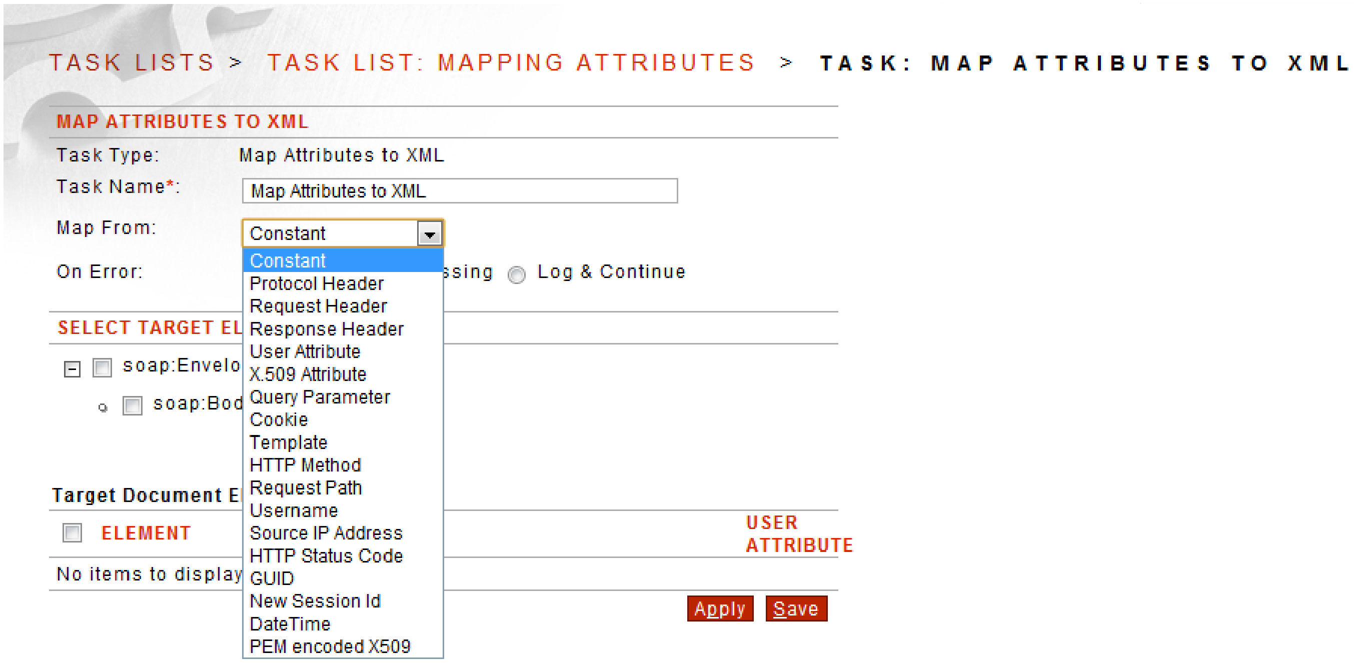
MAPPING

Source Type:	Request Header
Source Name*:	
Target Type:	User Attribute
Target Name*:	

Map Attributes and Headers Screen Terms

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.



TASK LISTS > TASK LIST: MAPPING ATTRIBUTES > TASK: MAP ATTRIBUTES TO XML

MAP ATTRIBUTES TO XML

Task Type:	Map Attributes to XML
Task Name*:	Map Attributes to XML
Map From:	Constant
On Error:	ssing ○ Log & Continue

SELECT TARGET EL

soap:Envelo

soap:Bod

Target Document E

□ ELEMENT	USER ATTRIBUTE
No items to display	

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.
- **X509 Attribute**
    - OID and other attributes within an X509 cetificate
- **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
- **HTTP Method**
    - GET or POST
- **Request Path**
    - The request path of the inbound client request, not including 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
- **GUID**
    - A unique id generated to uniquely identify clients
- **New Session ID**
    - An id generated to uniquely identify sessions using basic HTTP authentication
- **DateTime**
    - Current Date and Time
- **PEM encoded X509**
    - PEM encoded X509 certificate

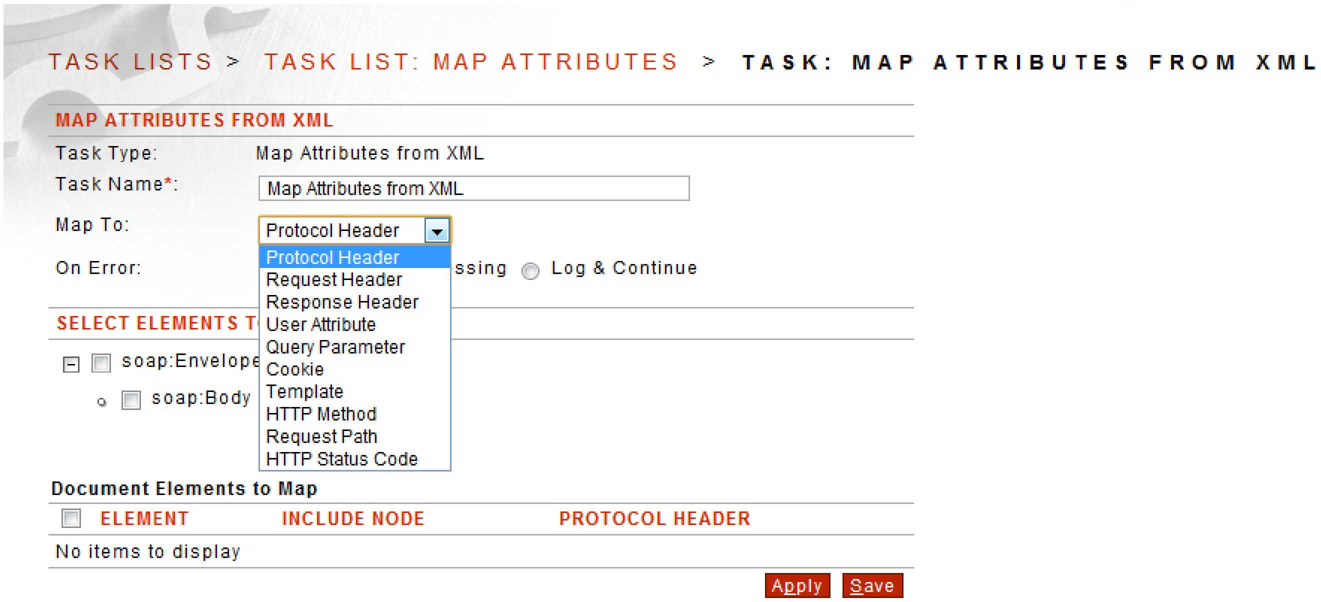
The Map Attribute to XML task provides the means to extract information in the form of attributes from various sources and map these values into the XML/SOAP document that is to be returned to the client, or proxied to the back-end system.

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 Attrbiutes 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: MAP ATTRIBUTES FROM XML

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



TASK LISTS > TASK LIST: MAP ATTRIBUTES > TASK: MAP ATTRIBUTES FROM XML

MAP ATTRIBUTES FROM XML

Task Type: Map Attributes from XML

Task Name*: Map Attributes from XML

Map To: Protocol Header

On Error: ssing ○ Log & Continue

SELECT ELEMENTS T

⊟ □ soap:Envelope

□ soap:Body

Document Elements to Map

□	ELEMENT	INCLUDE NODE	PROTOCOL HEADER
No items to display			

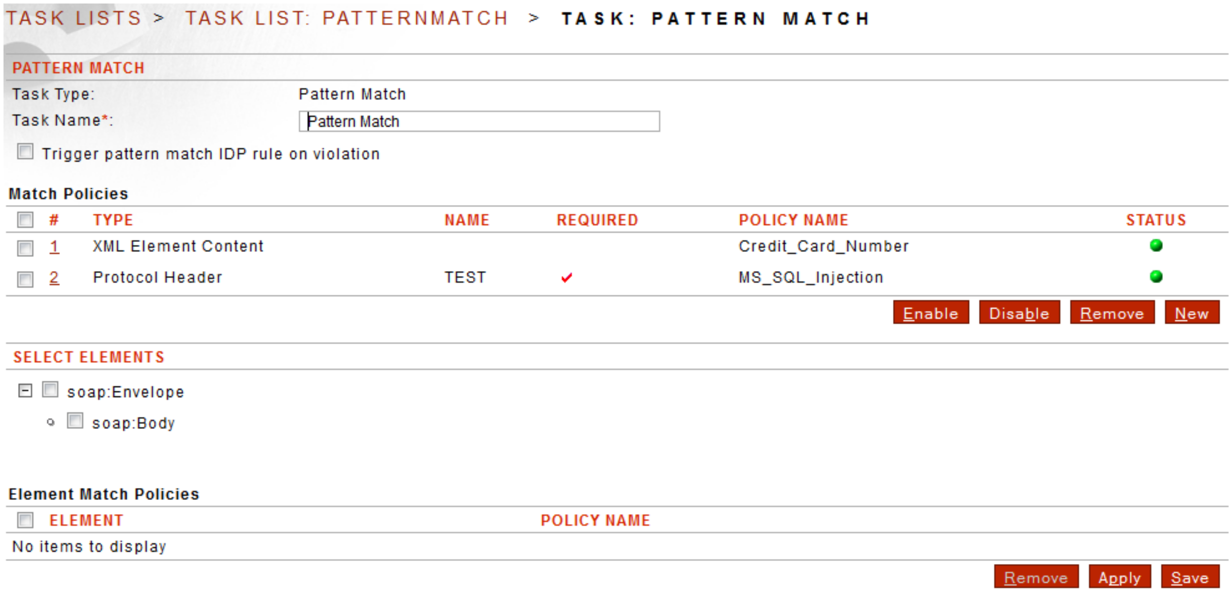
Map Attributes From XML Task Screen Terms

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 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

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*: Pattern Match

□ 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	●

SELECT ELEMENTS

⊟ □ soap:Envelope

○ □ soap:Body

Element Match Policies

□	ELEMENT	POLICY NAME
No items to display		

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:

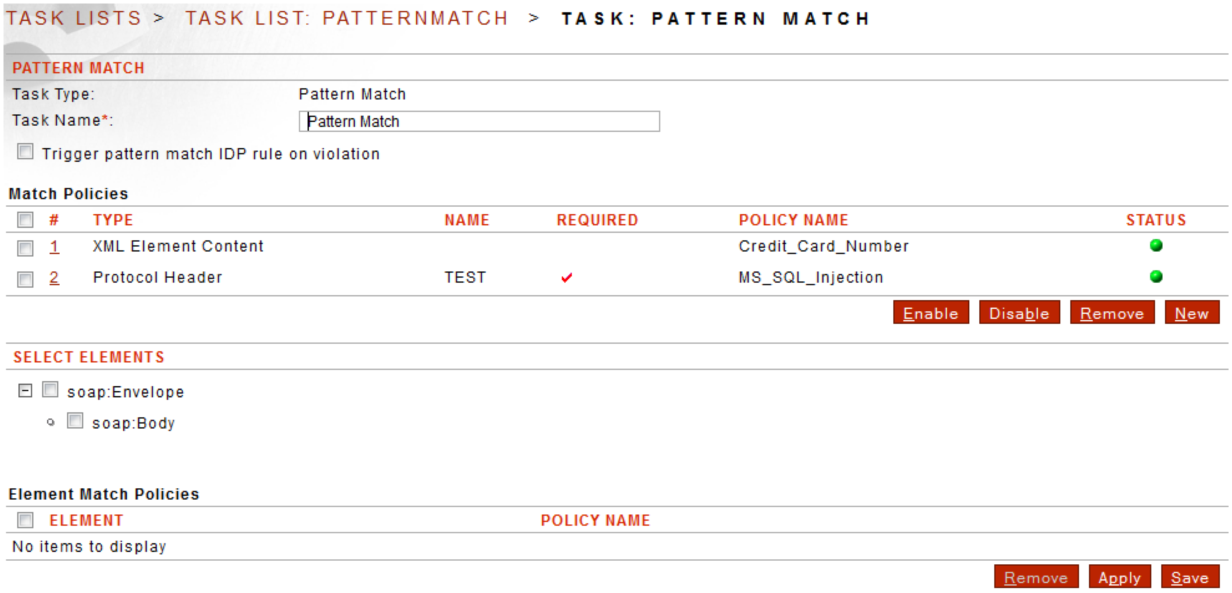
- **XML Element Content**
  - The targeted node value or attribute value from the select elements element match policy
- **Protocol Header**
  - If the task is operating on the request, this represents the request header. If the task is operating on the response, this setting represents the response 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

TASK: PATTERN MATCH

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

Pattern match task associates Pattern Policies, found under the Resources tab. Pattern Policies are sets of one or more Regular Expression policies that have been configured as ALLOW or DENY.



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

PATTERN MATCH

Task Type:	Pattern Match
Task Name*:	Pattern Match

□ 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	●

SELECT ELEMENTS

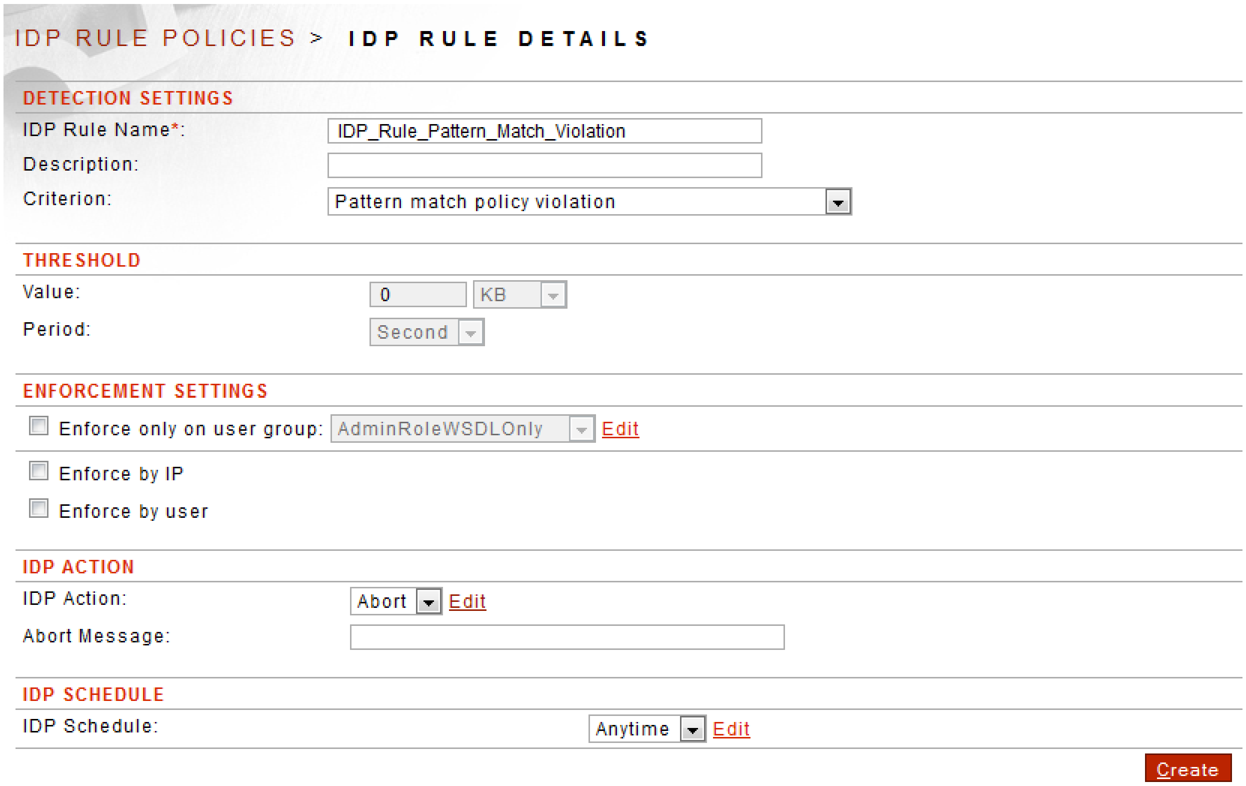
⊟ □ soap:Envelope
○ □ soap:Body

Element Match Policies

□	ELEMENT	POLICY NAME
No items to display		

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

□ Enforce by IP

□ Enforce by user

IDP ACTION

IDP Action: Abort

Abort Message:

IDP SCHEDULE

IDP Schedule: Anytime

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
Trigger Pattern Match IDP Rule on Violation	If the Pattern Policy is set to ALLOW/DENY, this will be enforced based on the affiliated IDP Rule “Pattern Match Policy Violation”
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:

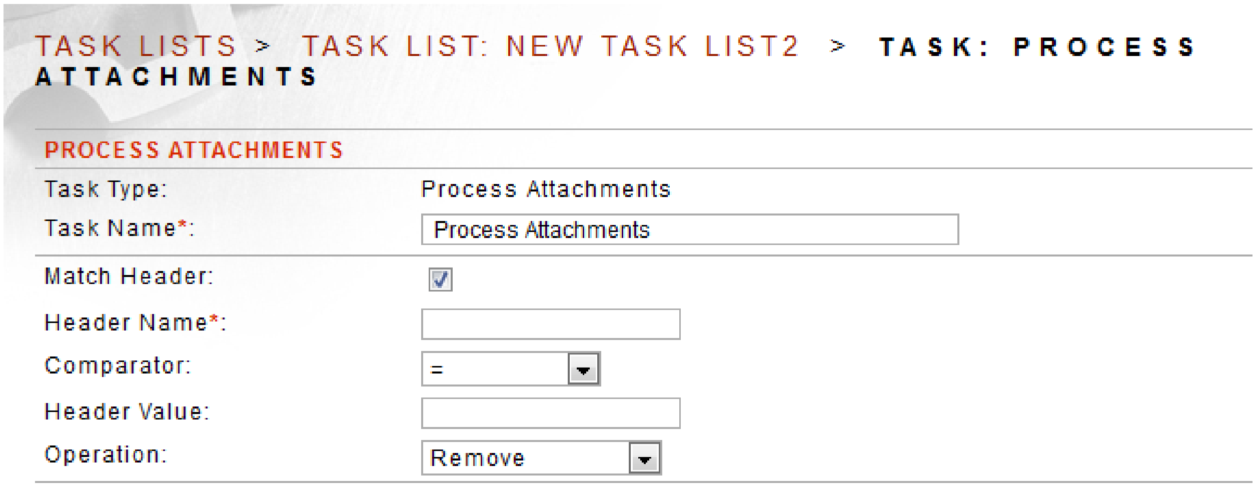
- **XML Element Content**
  - The targeted node value or attribute value from the select elements element match policy
- **Protocol Header**
  - If the task is operating on the request, this represents the request header. If the task is operating on the response, this setting represents the response 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

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

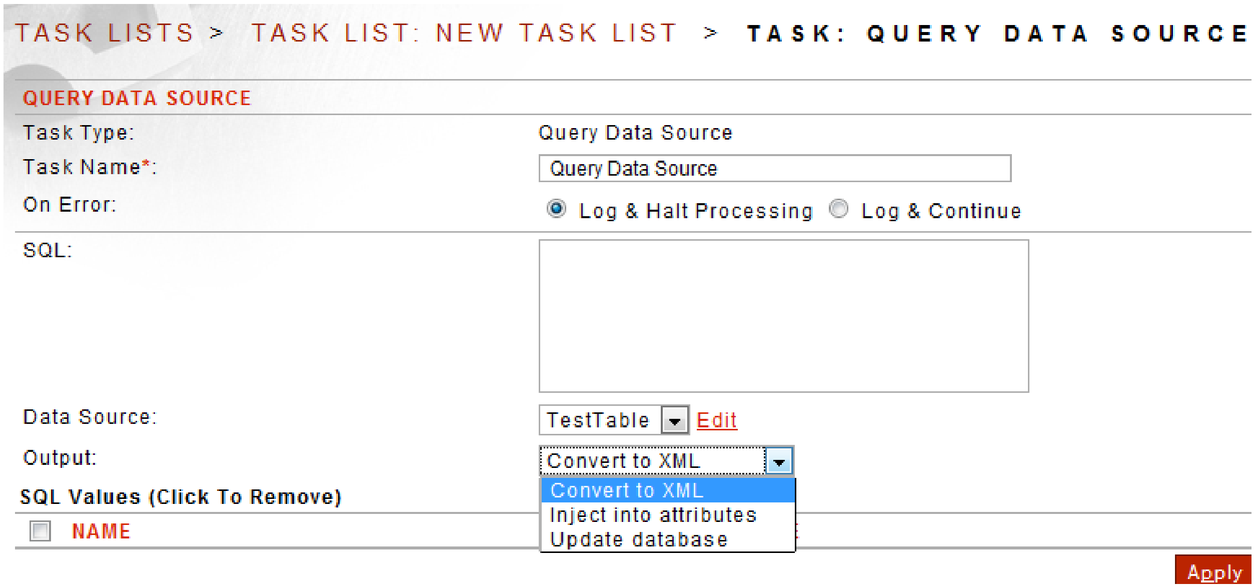
Process Attachments Task Screen Terms

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: • Remove: attachment will be stripped • Block: The request will be blocked due to the existence of the attachment • BASE64 Encode: Will encode the attachment with BASE64 encoding

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: NEW TASK LIST > TASK: QUERY DATA SOURCE

QUERY DATA SOURCE

Task Type: Query Data Source

Task Name*: Query Data Source

On Error: ● Log & Halt Processing ○ Log & Continue

SQL:

Data Source: TestTable Edit

Output: Convert to XML

SQL Values (Click To Remove)

□ NAME

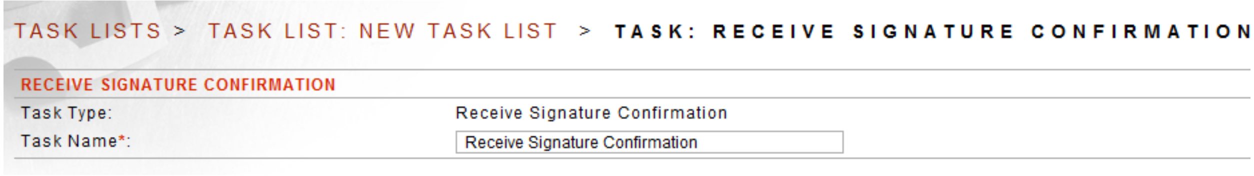
Query Data Source Task Screen Terms

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 substitiution 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	• **Convert to XML** o Automatically creates an XML document from the query response • **Inject Into Attributes** o 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. • **Update Database** o Optional setting that indicates that the SQL is an INSERT, DELETE, or UPDATE and does not return a value.

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

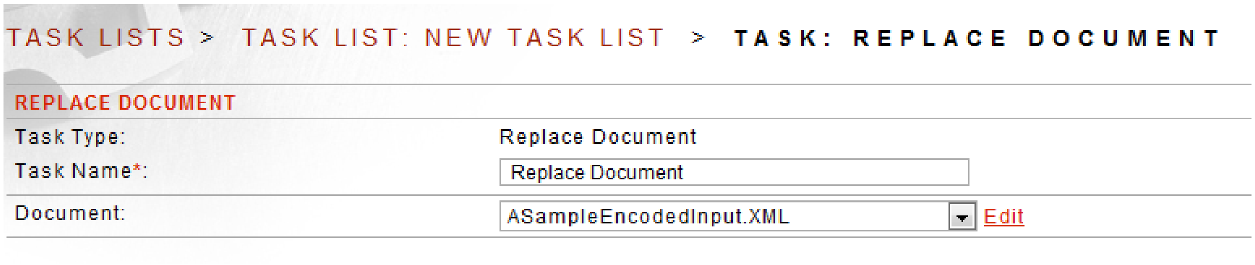
Task Type:	Receive Signature Confirmation
Task Name*:	Receive Signature Confirmation

Receive Signature ConfirmationTask Screen Terms

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

TASK: 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*:	Replace Document
Document:	ASampleEncodedInput.XML

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: 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:

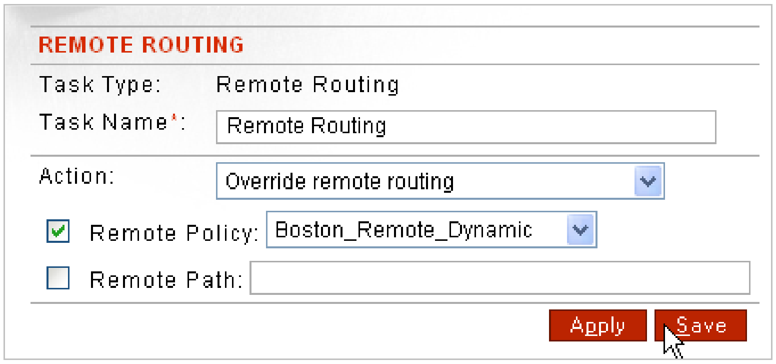
TERM	DEFINITION
Task Name	Identifier for this task.
Action	• With **Override remote routing** selected, the remote server for the request being processed is changed to the selected Remote Policy. Additionally, HTTP/S policies may override the Remote Path. • With **Send asynchronous message copy** selected, 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 asynchroneous response is received, it is logged but not used for further processing in the foreground Task List. • With **Send synchronous message copy** 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. • With **Replace message with remote response** 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.
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.

Remote Routing Task Examples

The example for the Remote Routing task is Add the “Remote Routing” Task to Route a Message to a Remote Policy Using an Existing Path.

Add the Remote Routing Task Using an Existing Path

Follow these steps to add the Remote Routing task to a Remote policy using an existing path. This instruction retains the physical path as specified.



REMOTE ROUTING

Task Type: Remote Routing

Task Name*: Remote Routing

Action: Override remote routing

☑ Remote Policy: Boston_Remote_Dynamic

□ Remote Path:

- From the TASK screen, select **New**.
- On the TASK TYPE screen, select the **Remote Routing** radio button, and then click **Next**.
- On the REMOTE ROUTING screen, select the **Override remote routing** option from the Action drop down list.
- Check the **Remote Policy** checkbox.

Note: Check the **Remote Policy** checkbox to retain the same remote path.

- From the Remote Policy drop down list, select a **Remote Policy** to use for re-routing.
- Skip the Remote Path field (optional) and then click **Save**.

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.

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

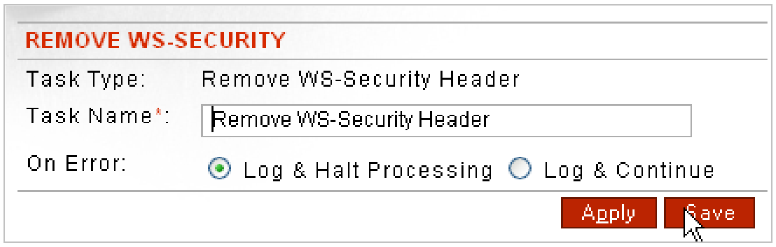
- OASIS WS-Security 1.1
- OASIS WS-Security 2004

Remove WS-Security Header Task Example

The example for the Remove WS-Security Header task is Add the "Remove a WS-Security Header" Task.

Add the Remove WS-Security Header Task

Follow these steps to add the “Remove a WS-Security header” task.



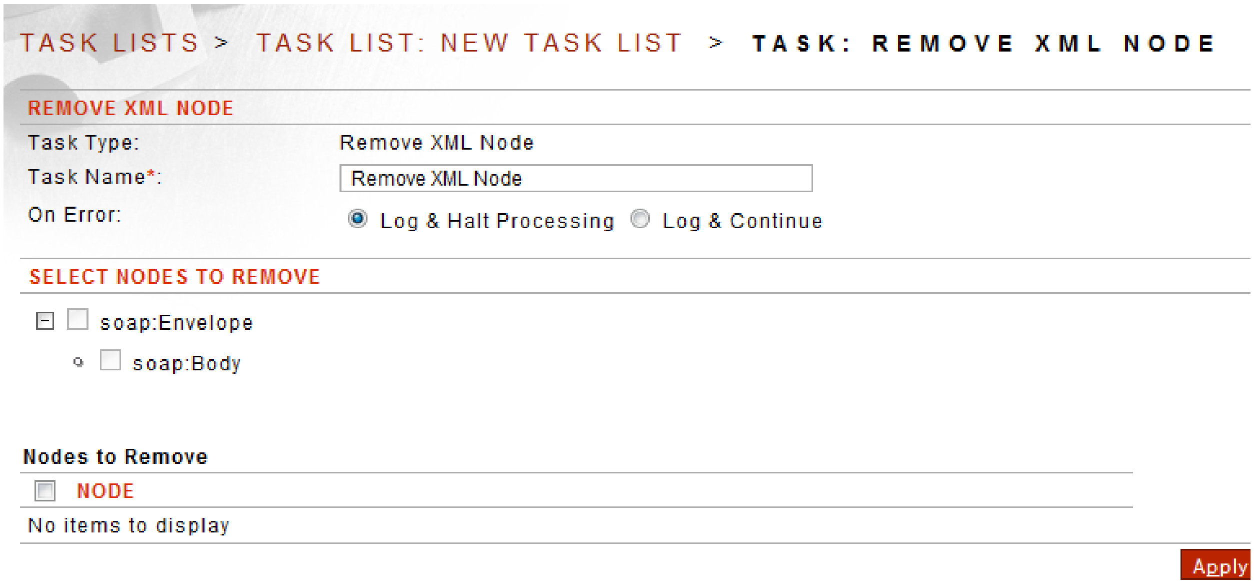
REMOVE WS-SECURITY

Task Type:	Remove WS-Security Header
Task Name*:	Remove WS-Security Header
On Error:	● Log & Halt Processing ○ Log & Continue

- From the TASKS screen, select New, and the TASK TYPE screen appears.
- Select the **Remove WS-Security Header** radio button, and then click **Next**.
- The REMOVE WS-SECURITY screen appears. Aligned with On Error, select the **Log and Halt** radio button.
- Select **Save**, and the TASK screen refreshes.

TASK: REMOVE XML NODE

The Remove XML Node task is used to remove elements from XML documents based on XPath expressions.



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

REMOVE XML NODE

Task Type: Remove XML Node

Task Name*: Remove XML Node

On Error: ● Log & Halt Processing ○ Log & Continue

SELECT NODES TO REMOVE

⊟ □ soap:Envelope

□ soap:Body

Nodes to Remove

□ NODE

No items to display

Replace Document 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: 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	• With the SAML 1.1 radio button selected, a SAML assertion is generated according to the SAML 1.1 specification. • With the SAML 2.0 radio button selected, a SAML assertion is generated according to the SAML 2.0 specification.
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	• When checked, specifies that the assertion is to be used by the recipient one time only and should not be cached for later use. • When unchecked, allows caching, which may decrease security.
Disallow reuse of this assertion	• When checked, enables SAML replay detection and only allows an assertion to be used once. • When unchecked, the SAML Assertion may be used more than once. SAML replay detection is disabled.
Identification Format	• With Email checked, identifies SAML Email token. The email identification format supports local and LDAP users only. • With X.509 Distinguished Name checked, identifies SAML X.509 DN token.

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	Specifies the statement type of SAML assertion to generate. Statement types are not mutually exclusive. • Authentication - Asserts that the user is authenticated and records the type of authentication used. • Attribute - Associates specified attributes with the user. • Authorization - Grants / denies the user access to a specified resource.
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	• With Username selected, an attribute with the value of the user name is included in the assertion. • With Email selected, an attribute with the value of the user email address is included in the assertion. • With DN selected, an attribute with the value of the user DN is included in the assertion. • 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. • 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. **Note:** For information on generating SAML attributes for Kerberos users, refer to the *Forum Systems Sentry™ Version 8.1 Kerberos Integration Guide*. • 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.

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	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:rwedc-negation”, then the user could type in one or more of the following values for the action: • Read - The subject may read the resource. • Write - The subject may modify the resource. • Execute - The subject may execute the resource. • Delete - The subject may delete the resource. • Control - The subject may specify the access control policy for the resource. 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. A SAML authority MUST NOT authorize both an action and its negated form.

SAML Assertion Task Examples

Examples for SAML Assertion task include:

- Add a SAML Assertion Email Token.

Note: For information on Kerberos tokens or adding a SAML Assertion with SAML custom Attribute from LDAP, refer to the *Forum Systems Sentry™ Version 8.1 Kerberos Integration Guide*.

For information on adding a SAML Assertion with SAML custom Attribute from cookie, refer to the *Forum Systems Sentry™ Version 8.1 CA™ SiteMinder APS Integration Guide*.

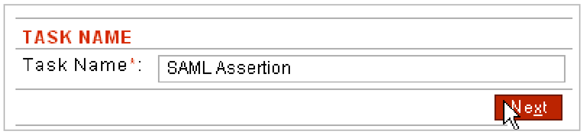
Add a SAML Assertion Email Token

Follow these steps to add a SAML Assertion Email Token to a message that did not previously have one.

SAML allows you to state that this individual has been authenticated at this particular time, for this particular duration of time, on this particular document. When the SAML Assertion is generated, the SAML Assertion based on the User's email is embedded in the document.

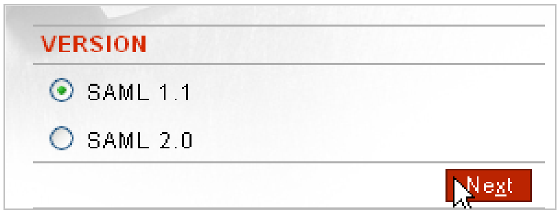
Use Case

Add an Email Token inside a SAML Assertion for Jack Kantos, someone in whom you have implicit trust. This instruction uses SoapDocument.xml as the sample XML document loaded in the Task List. This file is available under the Samples directory of the supplied CD. The system generates a token during this instruction.



TASK NAME

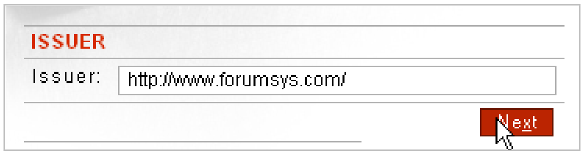
Task Name*: SAML Assertion



VERSION

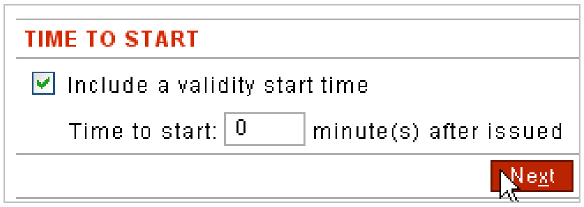
● SAML 1.1

○ SAML 2.0



ISSUER

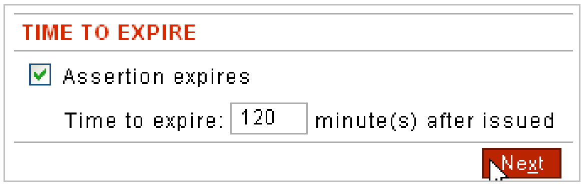
Issuer: http://www.forumsys.com/



TIME TO START

☑ Include a validity start time

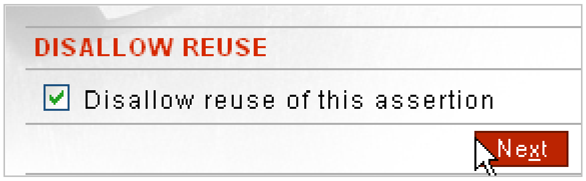
Time to start: 0 minute(s) after issued



TIME TO EXPIRE

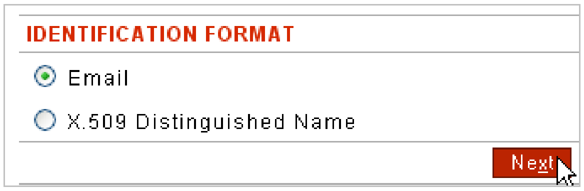
☑ Assertion expires

Time to expire: 120 minute(s) after issued



DISALLOW REUSE

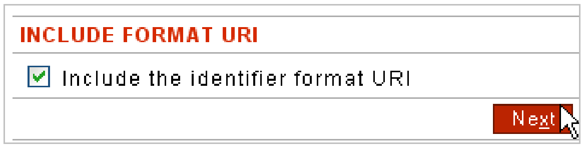
☑ Disallow reuse of this assertion



IDENTIFICATION FORMAT

● Email

○ X.509 Distinguished Name



INCLUDE FORMAT URI

☑ Include the identifier format URI

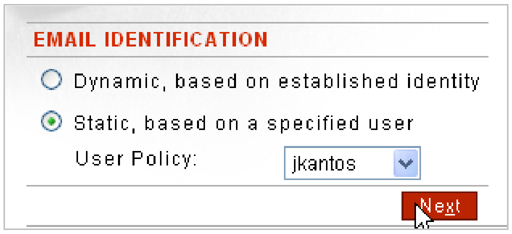
- From the **TASK LIST** screen, from the Sample Document drop down list, select **SoapDopcument.xml**.
- Select **New** and the TASK TYPE screen appears.
- Select the **SAML Assertion** radio button, and then click **Next**. The TASK NAME screen appears. Click **Next.**
- On the VERSION screen, select the **SAML 1.1** or **SAML 2.0** radio button, and then click **Next**.
- **ON the** ISSUER screen, accept the pre-populated value in the Issuer field. Click **Next.**
- The TIME TO START screen appears. Check the **Include a validity start time** checkbox, enter a **value** (**0**), and then click **Next**.

Note: The Time to start and time to expire may have 1 to 20 numeric characters. The default Time to start is 0 minutes, and the default Time to expire is 1 minute.

For designing real-time processing tasks, the Time to expire should reflect the smallest window of opportunity which allows SAML requests to pass through the product, as well as maintain the highest level of security.

For testing purposes, the Time to expire attribute should be increased, allowing time to complete testing and not allowing SAML assertions to expire. This is the reason to ignore processing errors.

- The TIME TO EXPIRE screen appears. Check the **Assertion expires** checkbox.
- In the Time to expire field, enter a **value** (**120**) as the time to expire after issued, and then click **Next**.
- The DISALLOW REUSE screen appears. Decide to check the **Disallow reuse of this assertion** or skip this option, and then click **Next**.
- The IDENTIFICATION FORMAT screen appears. Check the **Email** radio button, and then click **Next**.
- The INCLUDE FORMAT URI screen appears. Check the **Include the identifier format URI** checkbox, and then click **Next.**
- The EMAIL IDENTIFICATION screen appears. To configure the Token based on a specified user, click the **Static, based on a specified user** radio button. From the User policy drop down list, select a **User Policy** name, and then click **Next**.



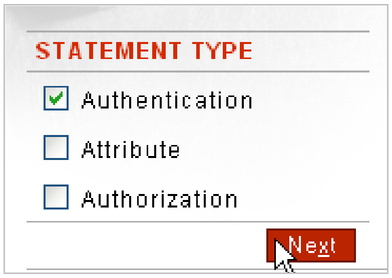
EMAIL IDENTIFICATION

○ Dynamic, based on established identity

● Static, based on a specified user

User Policy: jkantos

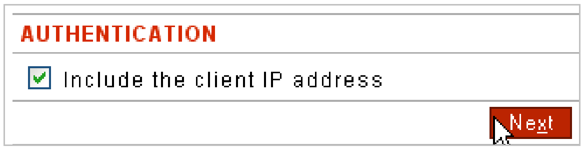
Note: Selecting the Static, based on a specified user radio button applies the selected Email Token to this SAML Assertion.


STATEMENT TYPE

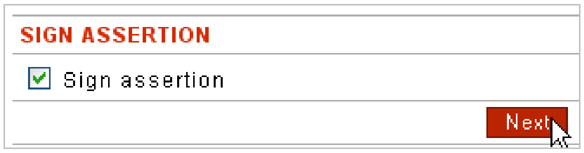
☑ Authentication

☐ Attribute

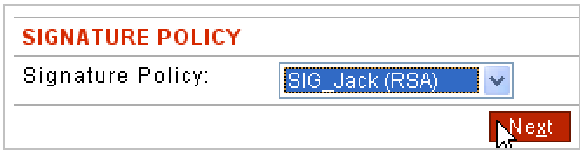
☐ Authorization


AUTHENTICATION

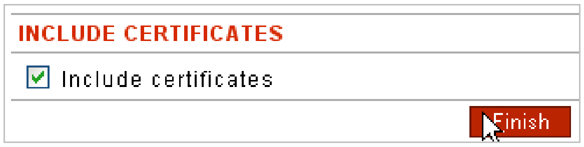
☑ Include the client IP address


SIGN ASSERTION

☑ Sign assertion


SIGNATURE POLICY

Signature Policy: SIG_Jack (RSA)

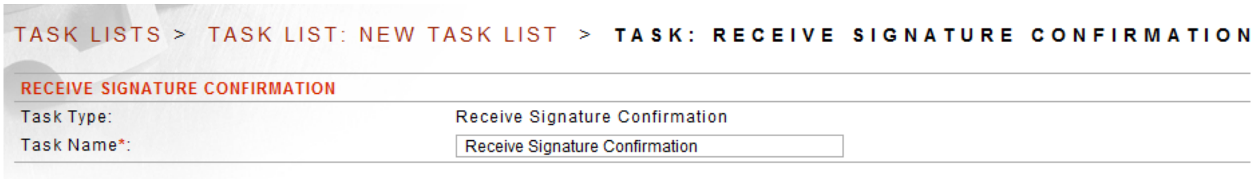

INCLUDE CERTIFICATES

☑ Include certificates

- The STATEMENT TYPE screen appears. Select the **Authentication** checkbox, and then click **Next**. The AUTHENTICATION screen appears.
- Check the **Include the client IP address** checkbox, and then click **Next**. The SIGN ASSERTION screen appears.
- Check the **Sign assertion** checkbox, and then click **Next**. The SIGNATURE POLICY screen appears.
- From the Signature Policy drop down list, select an **XML Signature policy name** (**SIG_JACK**), and then click **Next**. The INCLUDE CERTIFICATES screen appears.
- Check the **Include certificates** checkbox, and then click **Finish**. The TASKS screen refreshes.

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. ***Enveloped Signatures*** 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. ***Enveloping Signatures*** 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 to Web Services Security SOAP Messages with Attachments (SwA) Profile 1.1 specification. **Note:** When signing a Document with attachments, the signatures of the attachments are also inserted into the document. The attachments, themselves, are never modified.
Filter embedded content signatures (not recommended)	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. 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 Identifier	• 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. • With X.509 selected, the complete X.509 is used for identifying the key. • With SerialNumber selected, the X.509 issuer DN and serial number is used for identifying the key. • With SubjectKeyIdentifier selected, the X.509 v3 SubjectKeyIdentifier extension is used for identifying the key.
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

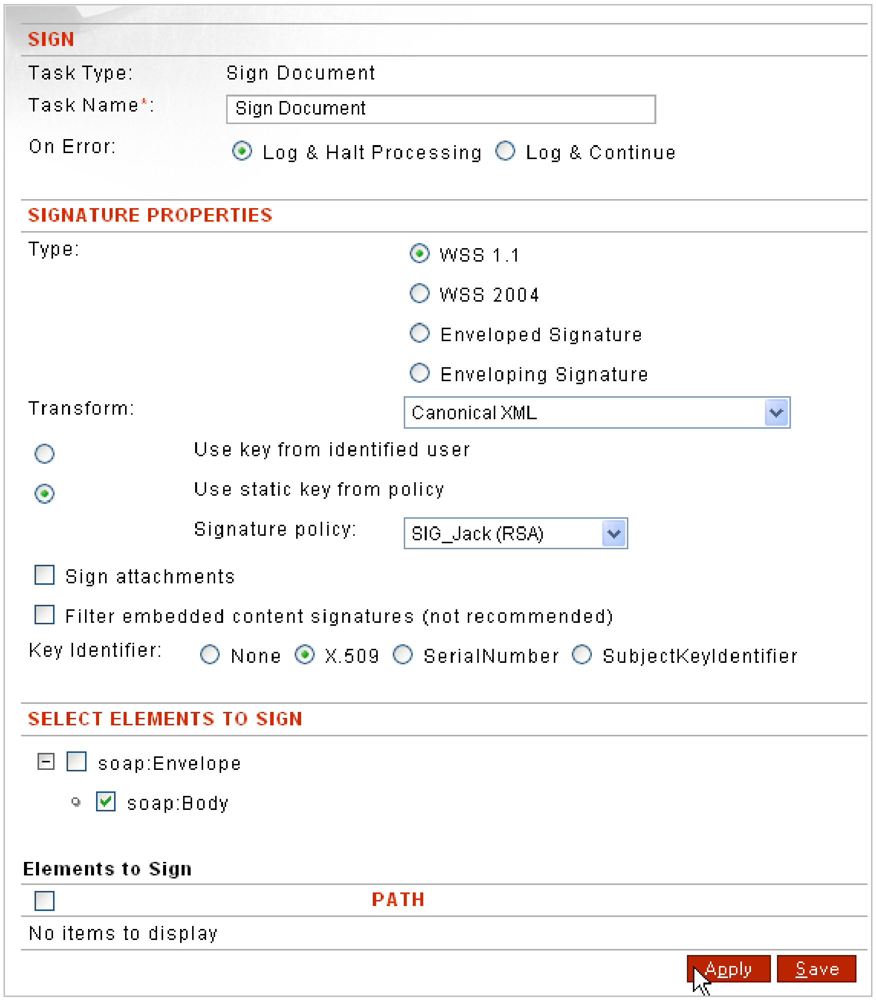
When adding an enveloped signature to the SOAP Envelope or SOAP Header of an incoming request that includes an ebXML MessageHeader SOAP header, the product detects the ebXML and applies an ebXML-compliant signature to the document. These actions are performed by the product automatically during the Sign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

Sign Document Task Examples

The example for the Sign Document tasks is Sign an Element of a Document.

Sign an Element of a Document

This instruction assumes you have configured the desired content filter during the Identify Document task and that an XML Signature policy has been created on the product. Follow these steps to sign an element of a document.



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

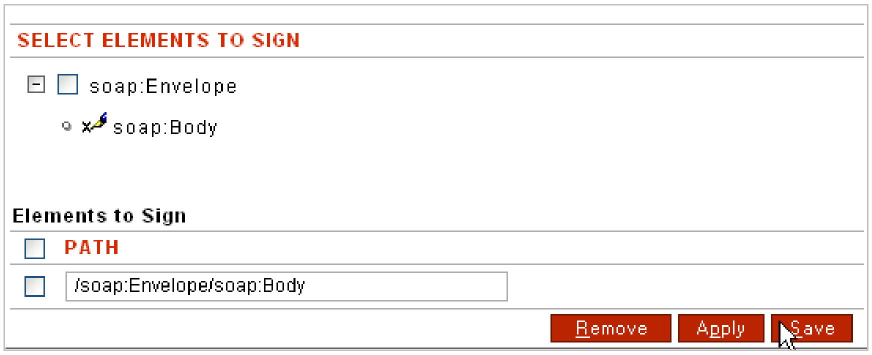
SELECT ELEMENTS TO SIGN

⊟ □ soap:Envelope

☑ soap:Body

Elements to Sign

□	PATH
No items to display	



SELECT ELEMENTS TO SIGN

⊟ □ soap:Envelope

soap:Body

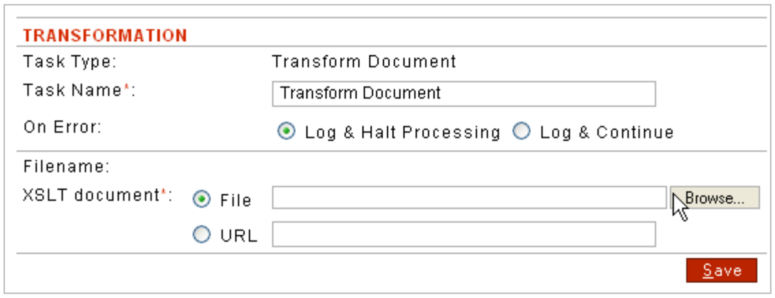
Elements to Sign

□	PATH
□	/soap:Envelope/soap:Body

- From the TASK screen, select **New**.
- On the TASK TYPE screen, select the **Sign Document** radio button, and then click **Next**.
- On the SIGN screen, aligned with On Error, select the **Log and Halt Processing** radio button.
- Aligned with Type, decide which signature type option to use and select that **option's** radio button.
- From the Transform drop down list, retain the Canonical XML option.
- Skip the Use key from identified user.
- Select the **Use static key from policy** radio button.
- From the Signature Policy drop down list, select an **XML Signature Policy**.
- Skip the Sign attachments checkbox.
- Skip the Filter embedded content signatures (not recommended) checkbox.
- Aligned with Key Identifier, select a **Key Identifier** option radio button.
- From the SELECT ELEMENTS TO SIGN section, check the **checkbox** prefacing the element to sign, and then select **Apply**. The SIGN screen refreshes, and the signed element is now prefaced by the Pen and X icon and the Elements to Sign area includes the path for the signed element.
- Select **Save**.

TASK: TRANSFORM DOCUMENT (XSLT)

The Transform Document task uses simple or compound XSLT 1.0 definitions to transform the target request or response document. XSLT 1.0 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	

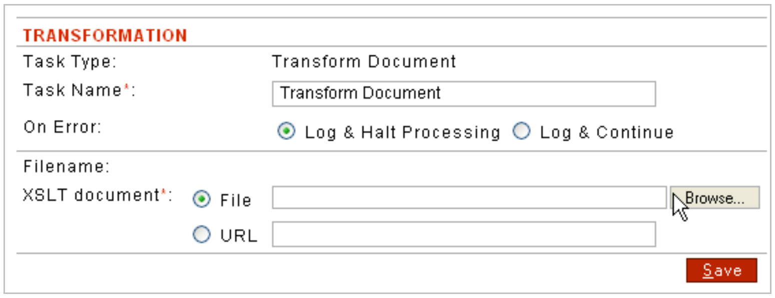
Tranform Document Task Screen Terms

TERM	DESCRIPTION OF OPTIONS
Task Name	The name of the task
XSLT Document	The simple or combound set of XSLT 1.0 documents that are to be used to transform the target document.

Transform XSLT Example

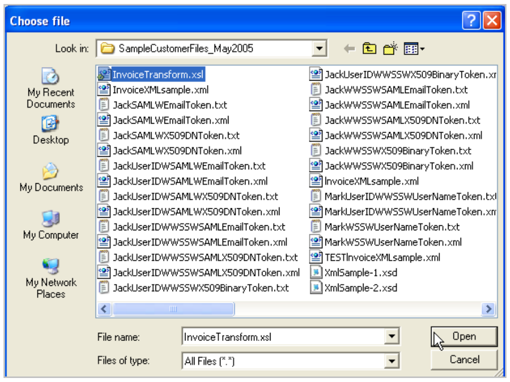
During the Transform Document task, the system applies an XSLT style sheet to the target document. This style sheet manipulates the data and transforms it per the XSLT definitions.

To set up a Transform Document task:



TRANSFORMATION

Task Type:		Transform Document
Task Name*:		Transform Document
On Error:		● Log & Halt Processing ○ Log & Continue
Filename:		
XSLT document*:	● File	
	○ URL	



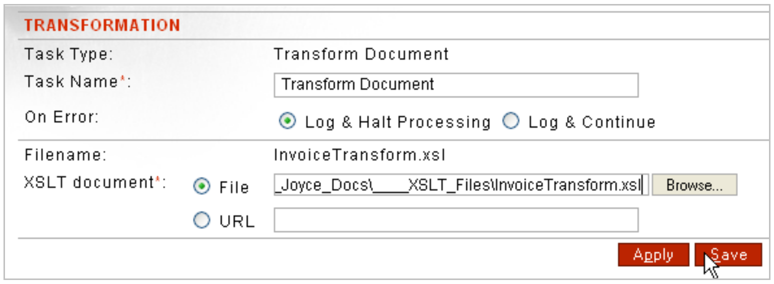
Choose file

Look in: SampleCustomerFiles_May2005

File name: InvoiceTransform.xsl

Files of type: All Files (*.*)

My Recent Documents
Desktop
My Documents
My Computer
My Network Places
InvoiceTransform.xsl
InvoiceXMLsample.xml
JackSAMLWEmailToken.txt
JackSAMLWEmailToken.xml
JackSAMLWX509DNToken.txt
JackSAMLWX509DNToken.xml
JackUserIDWSAMLWEmailToken.txt
JackUserIDWSAMLWEmailToken.xml
JackUserIDWSAMLWX509DNToken.txt
JackUserIDWSAMLWX509DNToken.xml
JackUserIDWWSSWSAMLEmailToken.txt
JackUserIDWWSSWSAMLEmailToken.xml
JackUserIDWWSSWSAMLX509DNToken.txt
JackUserIDWWSSWSAMLX509DNToken.xml
JackUserIDWWSSWX509BinaryToken.txt
JackUserIDWWSSWX509BinaryToken.xr
JackWWSSWSAMLEmailToken.txt
JackWWSSWSAMLEmailToken.xml
JackWWSSWSAMLX509DNToken.txt
JackWWSSWSAMLX509DNToken.xml
JackWWSSWX509BinaryToken.txt
JackWWSSWX509BinaryToken.xml
InvoiceXMLsample.xml
MarkUserIDWWSSWUserNameToken.txt
MarkUserIDWWSSWUserNameToken.xm
MarkWSSWUserNameToken.txt
MarkWSSWUserNameToken.xml
TESTInvoiceXMLsample.xml
XmlSample-1.xsd
XmlSample-2.xsd
Open
Cancel



TRANSFORMATION

Task Type: Transform Document

Task Name*: Transform Document

On Error: ● Log & Halt Processing ○ Log & Continue

Filename: InvoiceTransform.xsl

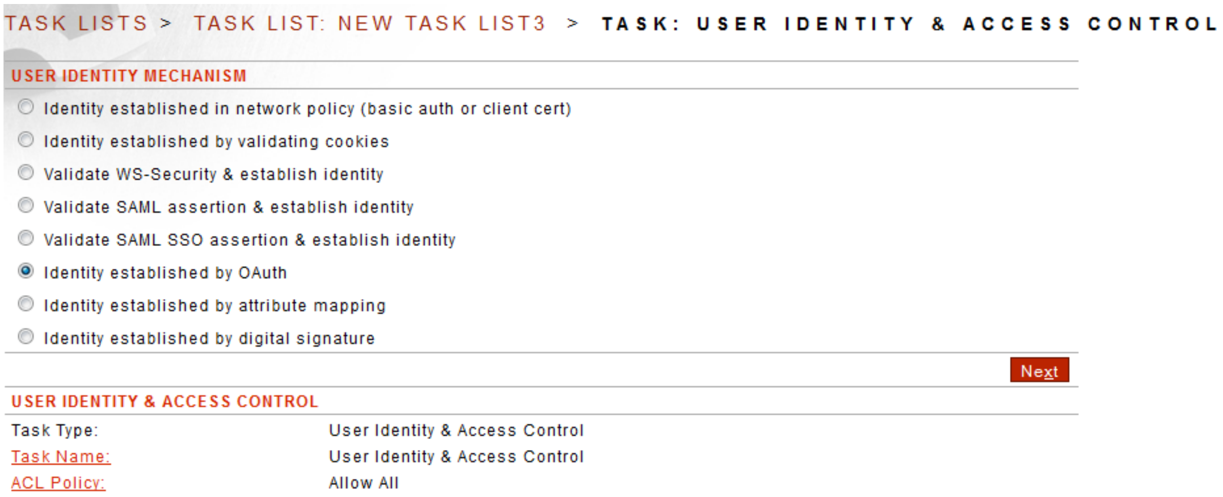
XSLT document*: ● File _Joyce_Docs____XSLT_Files\InvoiceTransform.xsl

○ URL

- From the TASK screen, select **New**, and the TASK TYPE screen appears.
- Select the **Transform Document** radio button, and then click **Next**.
- The TRANSFORMATION screen appears.
- Aligned with On Error, select the **Log and Halt Processing** radio button.
- Select the **Browse** radio button and the Choose file screen appears.
- Navigate to and highlight an **XSLT file** to import, and then click **Open**.
- The TRANSFORMATION screen refreshes.
- Select **Save** and the TASK screen refreshes.

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: NEW TASK LIST3 > 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
● Identity established by OAuth
○ Identity established by attribute mapping
○ Identity established by digital signature

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

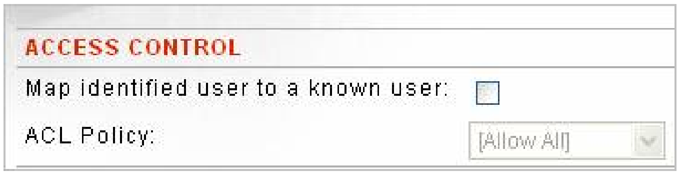
Access Control Lists (ACL) are sets of user groups which have either been granted or denied access to a Task List. The User Identity & Access Control task also allows Administrators to designate an Access Control List (ACL) for a given lists of Tasks, and establish an identity for the user. The identity 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.

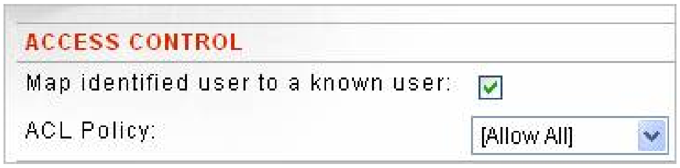


ACCESS CONTROL

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.

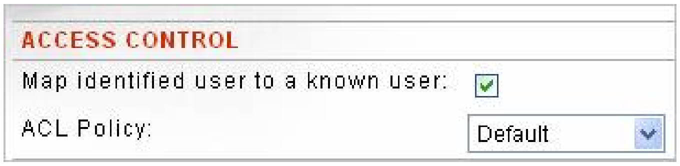


ACCESS CONTROL

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 8.1 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	• When checked, the user credentials are mapped to a known user configured in the system or in an external user store. • When unchecked, the user credentials are not mapped to a known user.
Access Control	The name of the access control list to apply to this task. For more information, refer to the Access Control Options with User Identity and Access Control tasks section discussed earlier in this chapter.
User Identity Mechanism	• With Identity established in network policy (password auth or client sert) selected, the user in the document is identified by protocol authentication such as HTTP Basic Auth or SSL. • With Validate WS-Security and establish identity selected, the user is identified from a security token in the WS-Security header. • With Validate SAML assertion and establish identity selected, the user in the document is identified from a SAML assertion. • 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 • With Identity established by OAuth, the OAuth credentials are extracted and validated • With Identity established by XML mapping selected, the username and password values entered in the Mapped Attributes dialog are used to identifiy the user in the document. • 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.
Security Token Type	• With Username token selected, a Username token is required in the document for user identification. • With X.509 binary token selected, an X.509 binary token is required in the document for user identification. • With SAML token selected, a SAML assertion required in the document for user identification. • With Kerberos token selected, a Kerberos token is required in the document for 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	• With Email selected, user identity is established by the Email address inside a SAML assertion. • With X.509 DN selected, user identity is established by the X.509 DN inside a SAML assertion. • With Attribute selected, user identity is established by the value of the attribute specified in the Attribute dialog.
Mapped Attributes	Username attribute is the attribute to which the username was mapped from the document in the proceeding Map Attributes from XML task. Password attribute is the attribute to which the optional password was mapped from the document in the proceeding Map Attributes from XML task.

Note: Terms that refer to SAML Assertions in the User Identity and Access Control Wizard Terms table can be found in the a section entitled SAML Assertion Task Terms.

User Identity and Access Control Task Examples

Examples for the User Identity and Access Control task, which is always performed dynamically, include:

- Protocol-based User Identity / Access Control.
- Add User Identity and Access Control by XML Mapping.
- Add User Identity and Access Control by Digital Signature.

Note: For information on Kerberos tokens or adding User Identity and Access Control using a Sample Document that Includes a Kerberos Token, refer to the *Forum Systems Sentry™ Version 8.1 Kerberos Integration Guide*.

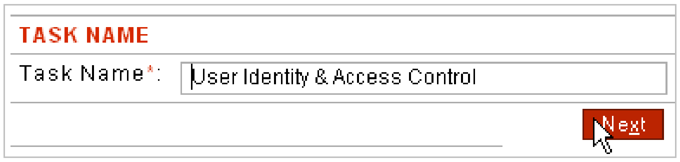
Protocol-based User Identity and Access Control

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.

USE CASE

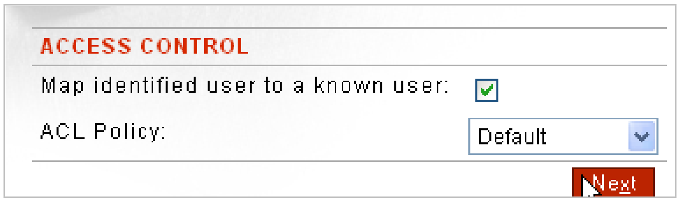
In order to authenticate a User whose identity will be communicated by HTTP Auth, the Server Policy will have to have authentication enabled so that the User Name / Password part of the HTTP session will be used by the User Identity & Access Control task. For testing purposes, when selecting Protocol-based identification, the Run and Settings commands prompt the User for User Name and Password. The Run dialogue is used to simulate a client web browser HTTP authentication.

Note: Before performing this operation, review the Prerequisites for All User Identity and Access Control Tasks listed earlier in this section.



TASK NAME

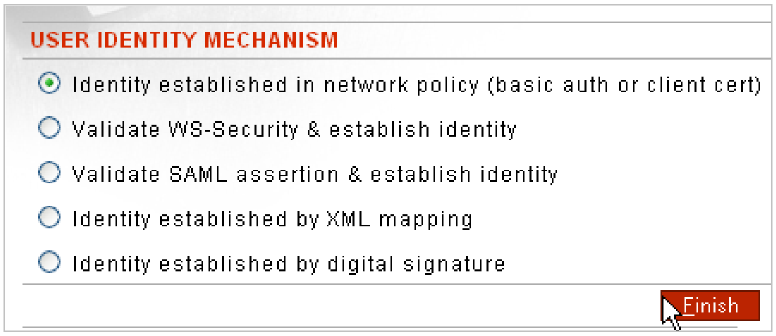
Task Name*: User Identity & Access Control



ACCESS CONTROL

Map identified user to a known user: ☑

ACL Policy: Default



USER IDENTITY MECHANISM

● Identity established in network policy (basic auth or client cert)

○ Validate WS-Security & establish identity

○ Validate SAML assertion & establish identity

○ Identity established by XML mapping

○ Identity established by digital signature

- 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 **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.

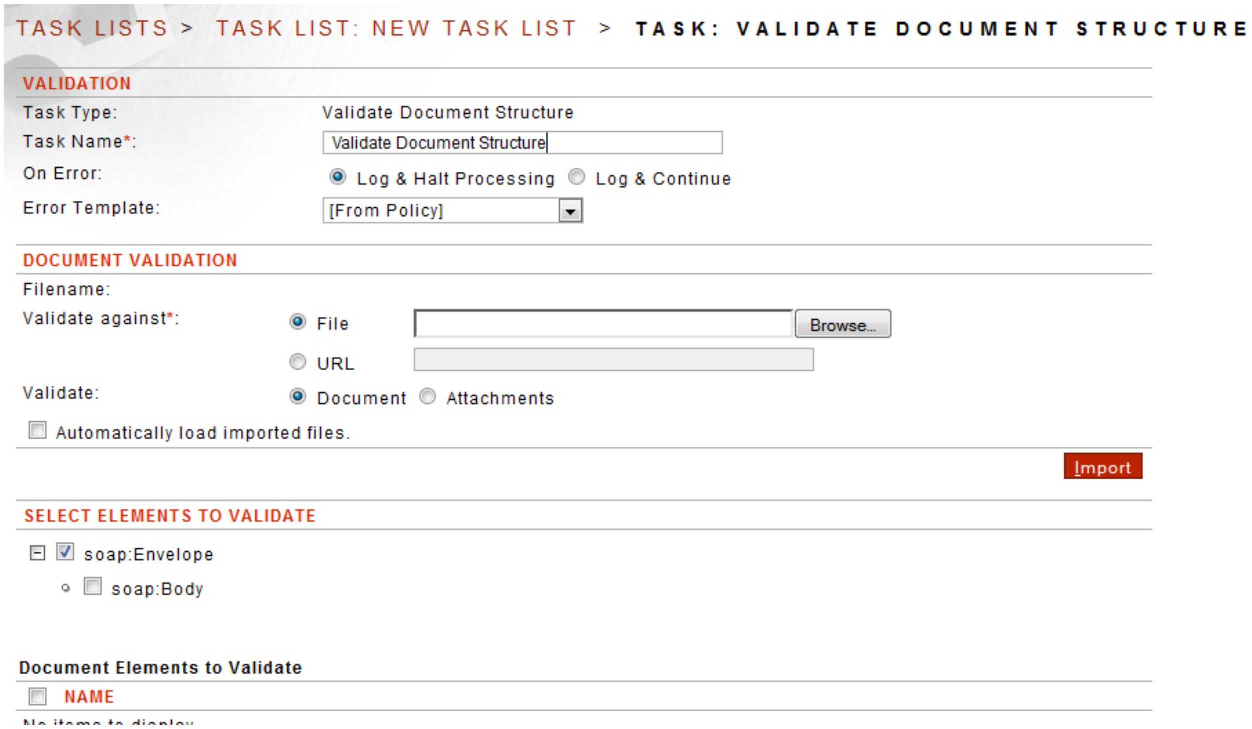
The actual attribute names used in this task can be anything as long as the same attribute names are specified in both tasks and the specified xml elements contain the actual username and password. The password may be omitted in both tasks if no password checking is required.

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.

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: NEW TASK LIST > TASK: VALIDATE DOCUMENT STRUCTURE

VALIDATION

Task Type: Validate Document Structure

Task Name*: Validate Document Structure

On Error: ● Log & Halt Processing ○ Log & Continue

Error Template: [From Policy]

DOCUMENT VALIDATION

Filename:

Validate against*: ● File ○ URL

Validate: ● Document ○ Attachments

□ Automatically load imported files.

SELECT ELEMENTS TO VALIDATE

☑ soap:Envelope

□ soap:Body

Document Elements to Validate

□ NAME

No items to display

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.

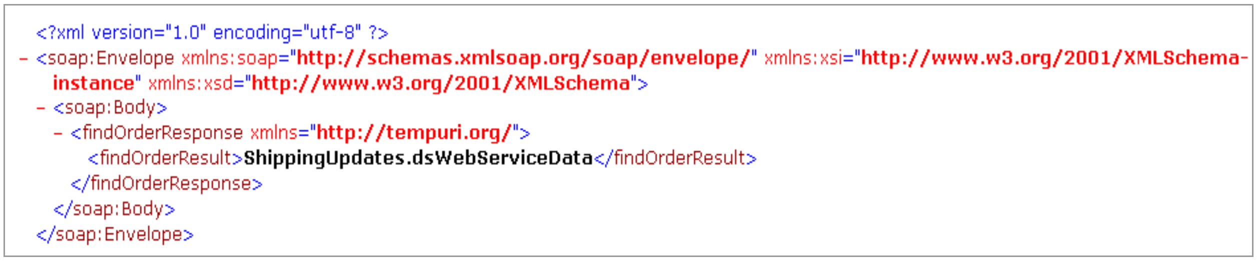
LAX Validation

The processContents=”lax” attribute can be used when only partial schema validation is desired, as in the following example from the SOAP 1.1 envelope schema:

```
   <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
processContents="lax" />
```

When the processContents=”lax” attribute appears on a schema element or attribute, the corresponding element or attribute will only be validated if the element or attribute matches the namespace of an available schema. If the appropriate schema for the lax content is unavailable, validation of that portion of the document will be skipped without error.

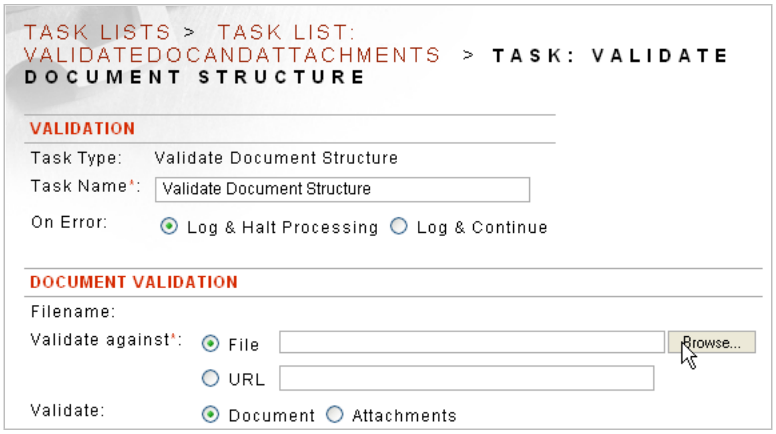
In the SOAP 1.1 envelope schema, the envelope, header, and body are set up with "lax" so that Administrators can put arbitrary XML content within those elements. The following graphic displays msnet.xml, which illustrates this. The <*findOrderResponse*> element is not in the SOAP schema, but since it is in the <soap:Body> which allows lax validation, the validation will succeed.



```
  <?xml version="1.0" encoding="utf-8" ?>
- <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  - <soap:Body>
    - <findOrderResponse xmlns="http://tempuri.org/">
        <findOrderResult>ShippingUpdates.dsWebServiceData</findOrderResult>
      </findOrderResponse>
    </soap:Body>
  </soap:Envelope>
```

Validate Document Structure with Schema

Follow these steps that will load an XML Schema, W3CSchema, XSD or DTD from a File.



TASK LISTS > TASK LIST: VALIDATEDOCANDATTACHMENTS > TASK: VALIDATE DOCUMENT STRUCTURE

VALIDATION

Task Type: Validate Document Structure

Task Name*: Validate Document Structure

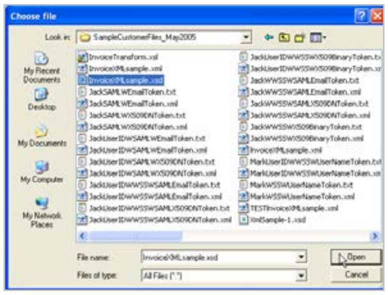
On Error: ● Log & Halt Processing ○ Log & Continue

DOCUMENT VALIDATION

Filename:

Validate against*: ● File ○ URL

Validate: ● Document ○ Attachments



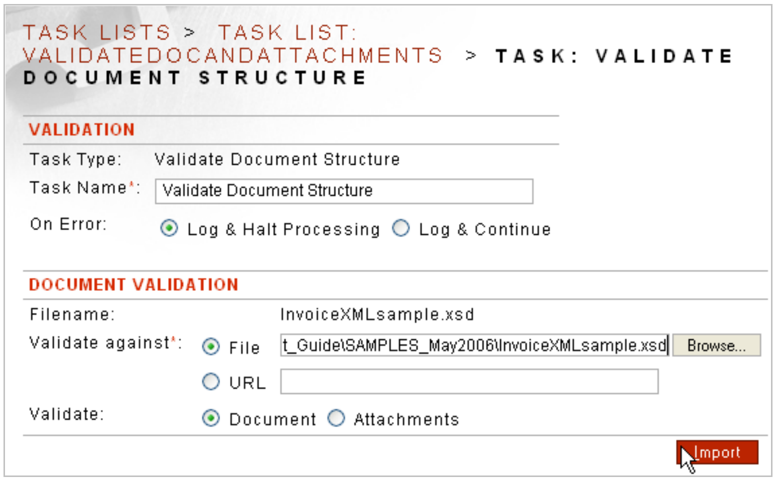
Choose file

Look in: SampleCustomerFiles_May2005

InvoiceTransform.xsl
InvoiceXMLsample.xml
InvoiceXMLsample.xsd
JackSAMLWEmailToken.txt
JackSAMLWEmailToken.xml
JackSAMLWX509DNToken.txt
JackSAMLWX509DNToken.xml
JackUserIDWSAMLWEmailToken.txt
JackUserIDWSAMLWEmailToken.xml
JackUserIDWSAMLWX509DNToken.txt
JackUserIDWSAMLWX509DNToken.xml
JackUserIDWWSSWSAMLEmailToken.txt
JackUserIDWWSSWSAMLEmailToken.xml
JackUserIDWWSSWSAMLX509DNToken.txt
JackUserIDWWSSWSAMLX509DNToken.xml
JackUserIDWWSSWX509BinaryToken.tx
JackUserIDWWSSWX509BinaryToken.x
JackWWSSWSAMLEmailToken.txt
JackWWSSWSAMLEmailToken.xml
JackWWSSWSAMLX509DNToken.txt
JackWWSSWSAMLX509DNToken.xml
JackWWSSWX509BinaryToken.txt
JackWWSSWX509BinaryToken.xml
InvoiceXMLsample.xml
MarkUserIDWWSSWUserNameToken.txt
MarkUserIDWWSSWUserNameToken.x
MarkWSSWUserNameToken.txt
MarkWSSWUserNameToken.xml
TESTInvoiceXMLsample.xml
XmlSample-1.xsd

File name: InvoiceXMLsample.xsd

Files of type: All Files (*.*)



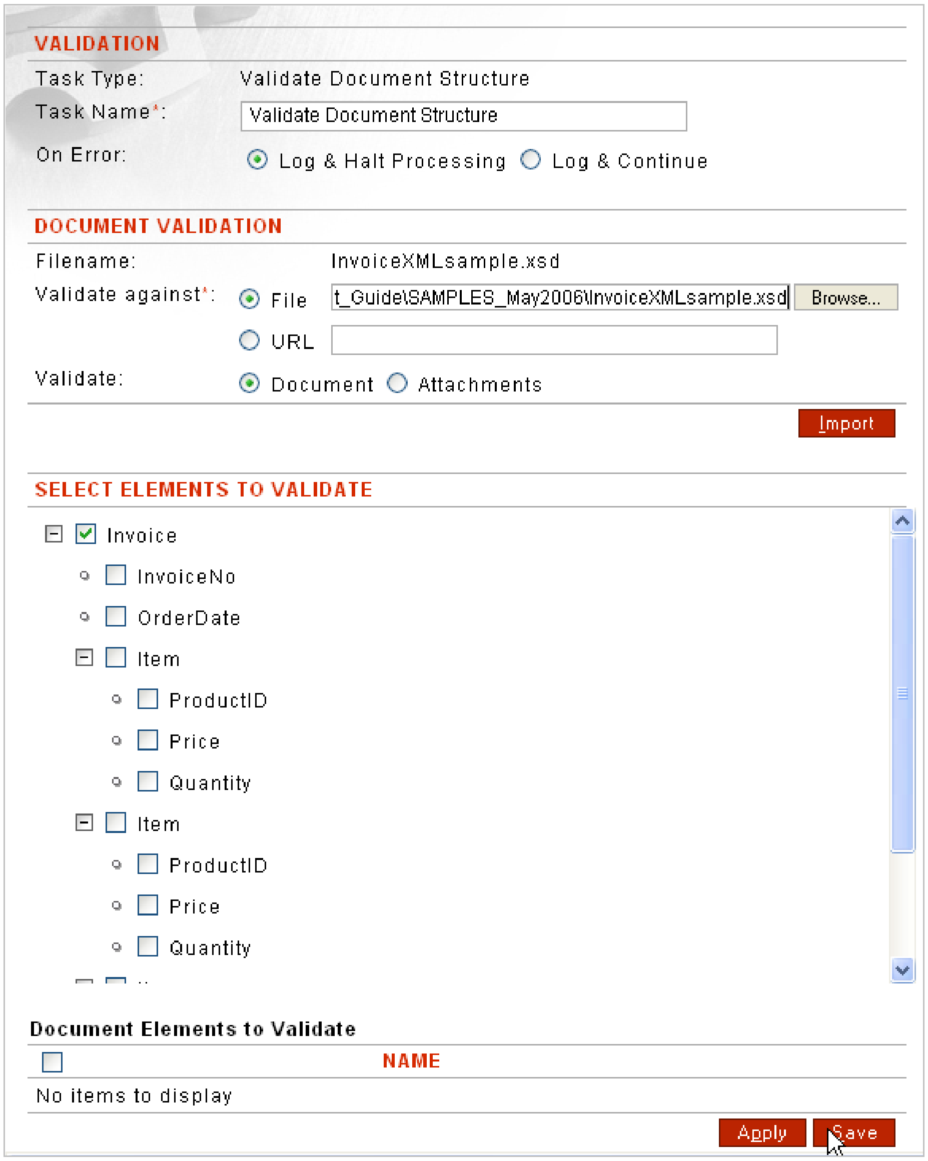
TASK LISTS > TASK LIST: VALIDATEDOCANDATTACHMENTS > TASK: VALIDATE DOCUMENT STRUCTURE

VALIDATION

Task Type:	Validate Document Structure
Task Name*:	Validate Document Structure
On Error:	● Log & Halt Processing ○ Log & Continue

DOCUMENT VALIDATION

Filename:		InvoiceXMLsample.xsd
Validate against*:	● File	t_Guide\SAMPLES_May2006\InvoiceXMLsample.xsd
	○ URL	
Validate:	● Document ○ Attachments	



VALIDATION

Task Type:	Validate Document Structure
Task Name*:	Validate Document Structure
On Error:	● Log & Halt Processing ○ Log & Continue

DOCUMENT VALIDATION

Filename:		InvoiceXMLsample.xsd
Validate against*:	● File	t_Guide\SAMPLES_May2006\InvoiceXMLsample.xsd
	○ URL	
Validate:	● Document ○ Attachments	

SELECT ELEMENTS TO VALIDATE

- ☑ Invoice
  - □ InvoiceNo
  - □ OrderDate
  - □ Item
    - □ ProductID
    - □ Price
    - □ Quantity
  - □ Item
    - □ ProductID
    - □ Price
    - □ Quantity

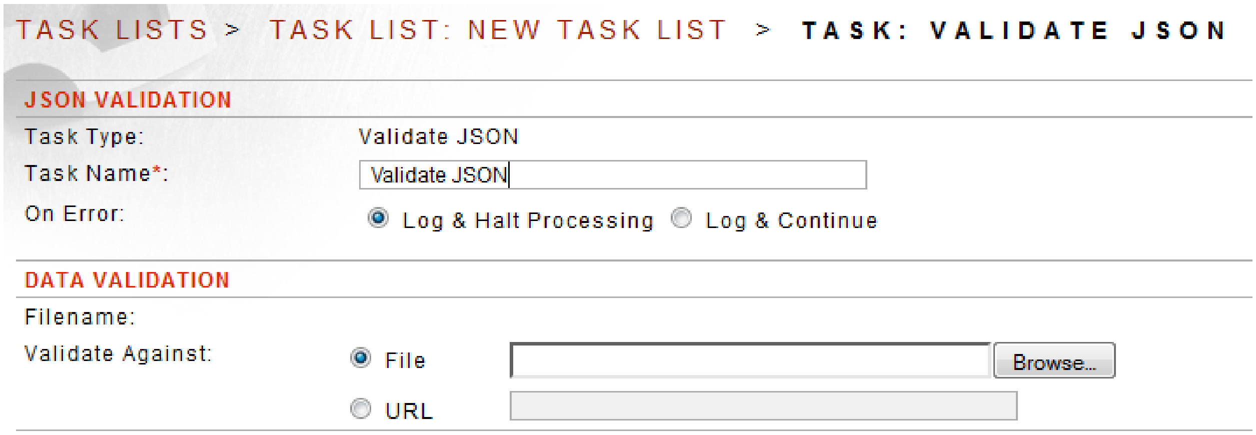
Document Elements to Validate

□	NAME
No items to display	

1. From the TASK screen, select **New.**
2. On the TASK TYPE screen, select the **Validate Document Structure** radio button, and then click **Next**.
3. On the VALIDATION screen, aligned with On Error, select the **Log & Halt Processing** radio button.
4. Aligned with Validate against, select the **File** radio button.
5. Confirm that the Validate document radio button is selected.
6. Click **Browse**, and then the Open screen appears.
7. From the Files of type drop down list, select **All Files**.
8. Navigate your local file system, locate and highlight an **XSD** or **DTD file**.
9. Click **Open**. The document name populates the Validate against field.
10. Click **IMPORT** and the VALIDATION screen refreshes with the SELECT ELEMENTS TO VALIDATE section populated with the root element in this document.
11. Select **Save**.

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: NEW TASK LIST > TASK: VALIDATE JSON

JSON VALIDATION

Task Type: Validate JSON

Task Name*: Validate JSON

On Error: ● Log & Halt Processing ○ Log & Continue

DATA VALIDATION

Filename:

Validate Against: ● File ○ URL

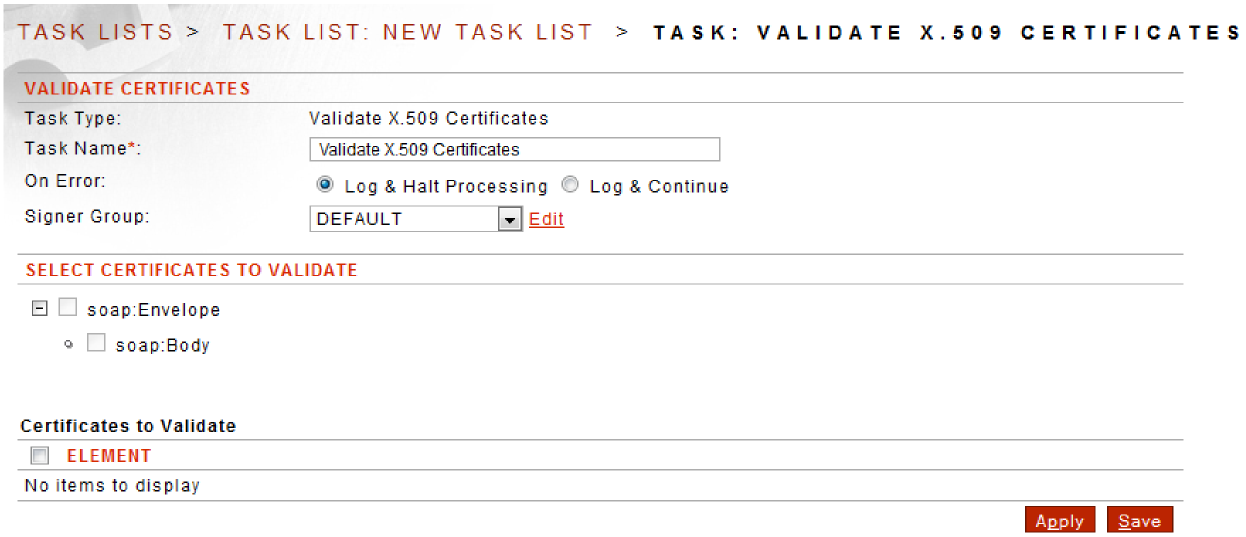
Validate JSON Task Screen Terms

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 Signauture 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.


TASK LISTS > TASK LIST: NEW TASK LIST > TASK: VALIDATE X.509 CERTIFICATES

VALIDATE CERTIFICATES

Task Type: Validate X.509 Certificates

Task Name*: Validate X.509 Certificates

On Error: ● Log & Halt Processing ○ Log & Continue

Signer Group: DEFAULT Edit

SELECT CERTIFICATES TO VALIDATE

□ soap:Envelope

□ soap:Body

Certificates to Validate

□	ELEMENT
No items to display	

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

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

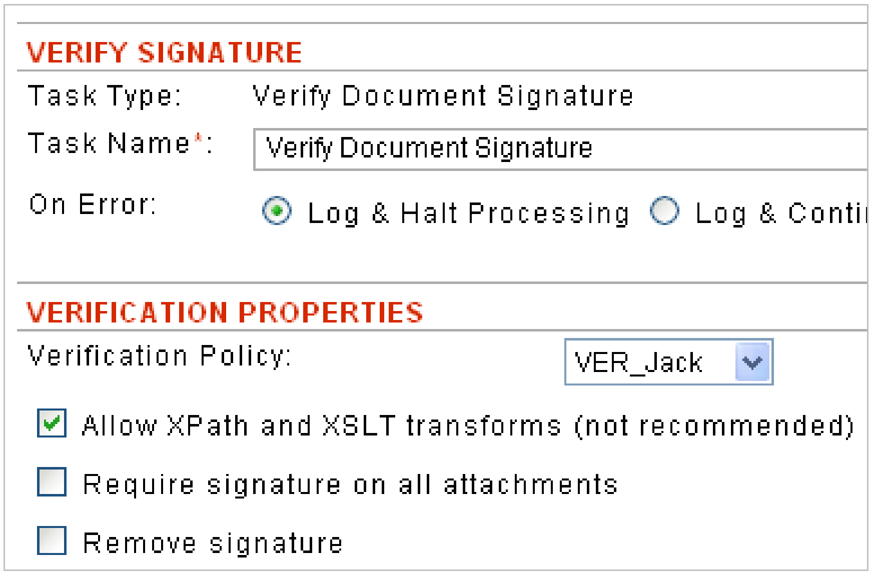
Signature Types Supported for Verification

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*: Verify Document Signature

On Error: ● Log & Halt Processing ○ Log & Conti

VERIFICATION PROPERTIES

Verification Policy: VER_Jack

☑ Allow XPath and XSLT transforms (not recommended)

□ Require signature on all attachments

□ Remove signature

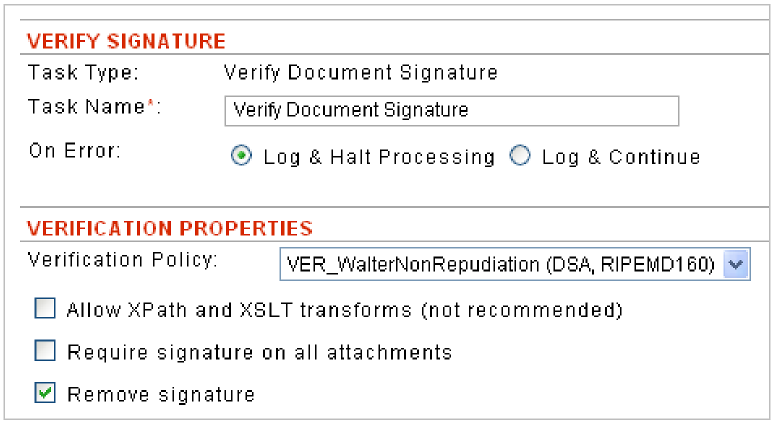
When ebXML signatures are present in the sample document, the **Allow XPath and XSLT Transforms** option is selected by default.

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*: Verify Document Signature

On Error: ● Log & Halt Processing ○ Log & Continue

VERIFICATION PROPERTIES

Verification Policy: VER_WalterNonRepudiation (DSA, RIPEMD160)

□ Allow XPath and XSLT transforms (not recommended)

□ Require signature on all attachments

☑ Remove signature

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

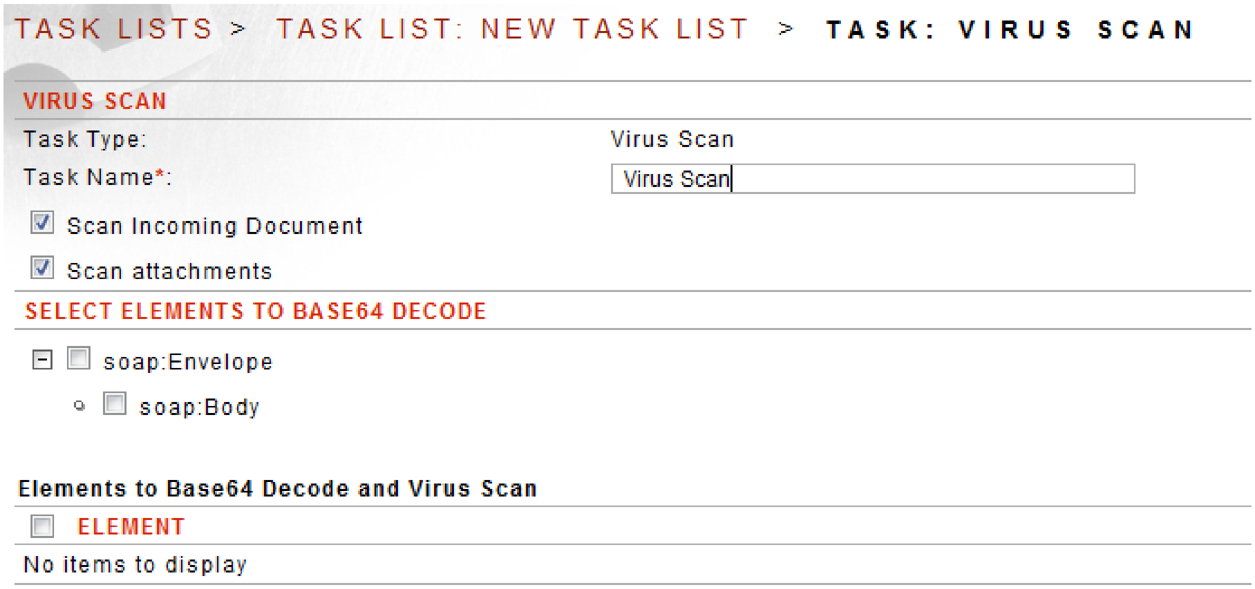
While verifying the signature on a document, please consider the following terms and definitions:

TERM	DESCRIPTION OF OPTIONS
VERIFY SIGNATURE	
On Error	• With Log & Halt Processing selected, if an error is encountered, the verification process will log an error and halt processing. • With Log & Continue selected, if an error is encountered, the verification process will log an error and continue processing.
VERIFICATION PROPERTIES	
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.
REQUIRED SIGNATURES	
Path	Node selected for verification options.
ELEMENTS REQUIRING SIGNATURE	
Path	Node that the verified signature must reference and include in the signed content

TASK: VIRUS SCAN

Forum Sentry is the industry only on-board virus scanning gateway that can scan for malware directly on-board with an integrated ClamAV virus engine. Malware threat vectors are scanned within XML data, as well as SOAP channels such as MIME, DIME, and MTOM, and HTML formats as well.

The integrated Sentry AV scanning engine is a highly optimized scanning engine capable of scanning documents up to 16GB in size.



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

VIRUS SCAN

Task Type: Virus Scan

Task Name*: Virus Scan

☑ Scan Incoming Document

☑ Scan attachments

SELECT ELEMENTS TO BASE64 DECODE

⊟ □ soap:Envelope

○ □ soap:Body

Elements to Base64 Decode and Virus Scan

□	ELEMENT
No items to display	

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- 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 ithe Maximum Clock Skew (in seconds) option on the Systems screen.

WS-Security Header with X.509 Token

The WS-Security Header includes the option to insert a X.509 Binary Token. However, simply sending X.509 Certificates with a request is not a sufficient method for authentication. Certificates are considered public knowledge, so anyone could include anyone else's Certificate with their request. The mechanism for using Certificates for authentication is based on signing some entity with the Private Key that corresponds to the Public Key in the Certificate.

In the case of sending a SOAP message, Forum Systems recommends that the Administrator create a digital signature of the SOAP body element with his/her Private Key, and then include the corresponding Certificate along with the signature in the headers of the request. Although including the Certificate with the request is not required, it does make it convenient for those trying to validate the signature.

WS-Security Header Task Options

The WS-Security Header task includes the following options that may or may not be selected regardless of which token the WS-Security Header consumes:

- WS-Security Configuration
- WS-Security Header mustUnderstand attribute.
- Message expiration.
- HTTP SOAP Action.

Each of these options is described next or in the WS-Security Header Task Wizard Terms section.

WS-Security Configuration

With a WS-Security Header task visible, select the WSS 2004 or WSS 1.1 radio button to configure the WS-Security Header according to the WSS 2004 or WSS 1.1 specifications.

WS-Security Header mustUnderstand Attribute

The WS-Security Header includes a mustUnderstand attribute. Mandatory processing means that the target web service must be WS-Security-aware.

If this default setting is not compatible with a particular web service implementation, the Administrator may uncheck the box, thus not requiring the consumption of the WSS header.

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	• When checked, WS-Security processing is mandatory
Must Understand checkbox	• 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. • When unchecked, WS-Security processing by the recipient is not mandatory.
SOAP Action	The HTTP SOAPAction field allows the user to specify the HTTP SOAPAction header value when the appliance creates a SOAP message, and applies only when the appliance receives a non-SOAP message and converts the message into SOAP format. The SOAPAction header is set to an empty string by default. If the default setting is not compatible with a particular web service implementation, the user may set the SOAPAction value to whatever is required by the web service implementation.
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	• With No Token selected, no security token is generated. • With Username token selected, a Username token is added to the document. • With X.509 binary token selected, an X.509 binary token is added to the document. • With SAML token selected, a SAML token is added to the document.
Password type	• With None selected, no password is selected. • With Clear Text selected, a password is included in cler text format. • With SHA 1 Digest selected, a SHA 1 digest of the password is included.
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.

TERM	DEFINITION
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	• 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. • Selecting the Static, based on specified user radio button adds the X.509 certificate of the specified user to this WS-Security Header.
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.

Note: Terms that refer to SAML Assertions in the WS-Security Header Task Wizard Terms table can be found in a previous section entitled SAML Assertion Task Terms.

WS-Security Header Task Examples

Examples for WS-Security Header task include:

- Add a WS-Security Header with User Name Token and Replay Verification.
- Add a WS-Security Header with X.509 Binary Token.
- Add a WS-Security Header with SAML Assertion X.509 Distinguished Name Token.
- Add WS-Security Header with SAML Assertion Custom Attribute from LDAP.

Note: For information on Kerberos tokens or adding a WS-Security Header with SAML Assertion Custom Attribute from LDAP, refer to the *Forum Systems Sentry™ Version 8.1 Kerberos Integration Guide*.

Add a WS-Security Header with User Name Token

Note: The system generates a token during this instruction.

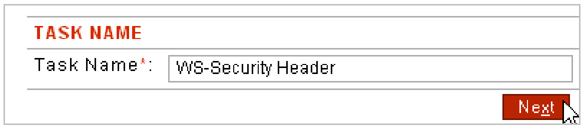
Follow these steps to add a WS-Security Header with a User Name Token to a message that did not previously have one.

USE CASE

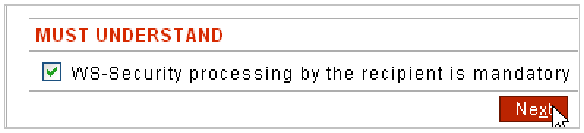
This instruction uses SoapDocument.xml as the sample XML document loaded in the Task List. This file is available under the Samples directory of the supplied CD.

This operation assumes that the user Mark Cross was created on the product with the option to save his password in clear text. This operation adds a User Name Token for Mark Cross inside a WS-Security Header. The resulting document will be wrapped in a SOAP envelope.

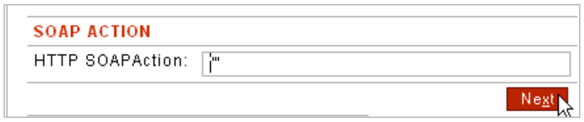
Note: Before performing this operation, review the Prerequisites for All WS-Security Tasks listed earlier in this section.


TASK NAME

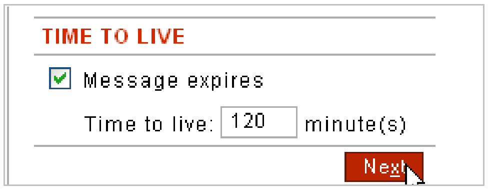
Task Name*: WS-Security Header


MUST UNDERSTAND

☑ WS-Security processing by the recipient is mandatory

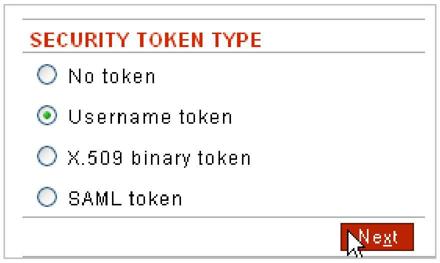

SOAP ACTION

HTTP SOAPAction: ""


TIME TO LIVE

☑ Message expires

Time to live: 120 minute(s)


SECURITY TOKEN TYPE

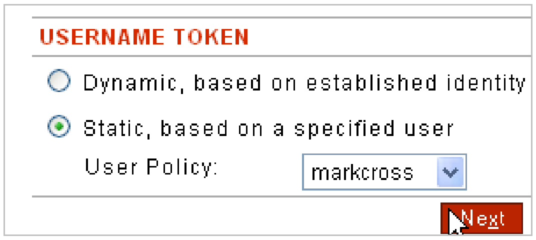
○ No token

● Username token

○ X.509 binary token

○ SAML token

- From the **TASK LIST** screen, from the Sample Document drop down list, select **SoapDopcument.xml**.
- Select **New** and the TASK TYPE screen appears.
- Select the **WS-Security Header** radio button, and then click **Next**.
- The TASK NAME screen appears. Click **Next**.
- The MUST UNDERSTAND screen appears. Check the **WS-Security processing by the recipient is mandatory** checkbox, and then click **Next**.
- The SOAP ACTION screen appears. Click **Next**.
- The TIME TO LIVE screen appears. Check the **Message expires** checkbox.
- Overwrite the **value** in the Time to live field (**120**) as the time to live after issued, and then click **Next**.
- The SECURITY TOKEN TYPE screen appears. Select the **Username token** radio button, and then click **Next**. The USERNAME TOKEN screen appears.


USERNAME TOKEN

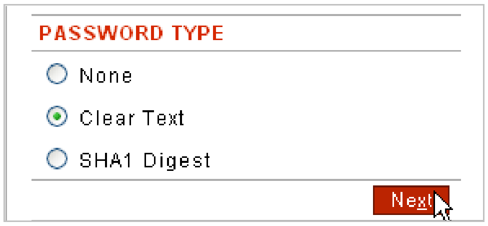
○ Dynamic, based on established identity

● Static, based on a specified user

User Policy: markcross

- To configure the Token based on a specified user, click the **Static, based on a specified user** radio button.

Note: Selecting the Static, Based on Specified User radio button applies the User Name Token of the specified user to this WS-Security Header.

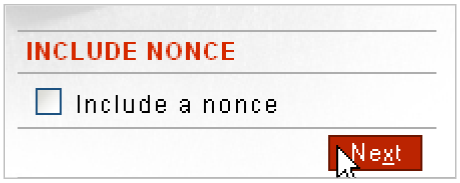


PASSWORD TYPE

○ None

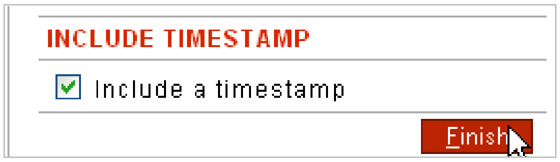
● Clear Text

○ SHA1 Digest



INCLUDE NONCE

□ Include a nonce



INCLUDE TIMESTAMP

☑ Include a timestamp

- From the User Policy drop down list, click a **User Name**, and then click **Next.**
- The PASSWORD TYPE screen appears. Select the **Clear Text** radio button, and then click **Next**.
- The INCLUDE NONCE screen appears. Skip the Include a nonce option, and then click **Next**.
- The INCLUDE TIMESTAMP screen appears. Check the **Include a timestamp** checkbox, and then click **Finish**.

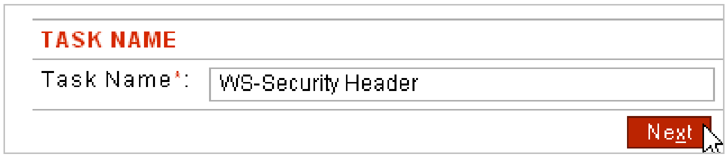
Add a WS-Security Header with X.509 Binary Token

Follow these steps to add a WS-Security Header with an X.509 Binary Token to a message that did not previously have one.

Use Case

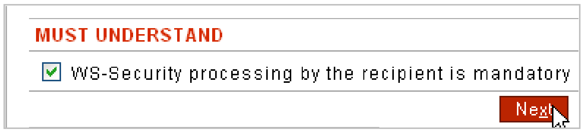
This instruction uses SoapDocument.xml as the sample XML document loaded in the Task List. This file is available under the Samples directory of the supplied CD. This operation adds an X.509 binary token inside a WS-Security Header. The resulting document will be wrapped in a SOAP envelope.

Note: Before performing this operation, review the Prerequisites for All WS-Security Tasks listed earlier in this section. The system generates a token during this instruction.



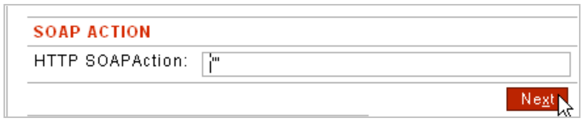
TASK NAME

Task Name*: WS-Security Header



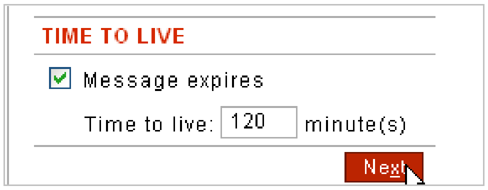
MUST UNDERSTAND

☑ WS-Security processing by the recipient is mandatory



SOAP ACTION

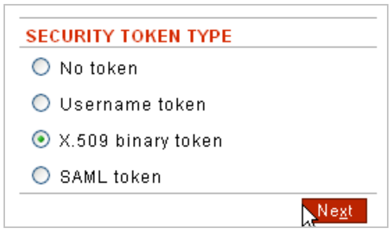
HTTP SOAPAction: ""



TIME TO LIVE

☑ Message expires

Time to live: 120 minute(s)



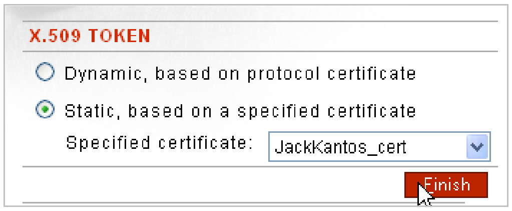
SECURITY TOKEN TYPE

○ No token

○ Username token

● X.509 binary token

○ SAML token



X.509 TOKEN

○ Dynamic, based on protocol certificate

● Static, based on a specified certificate

Specified certificate: JackKantos_cert

- From the **TASK LIST** screen, from the Sample Document drop down list, select **SoapDopcument.xml**.
- Select **New** and the TASK TYPE screen appears.
- Select the **WS-Security Header** radio button, and then click **Next**.
- The TASK NAME screen appears. Click **Next**.
- The MUST UNDERSTAND screen appears. Check the **WS-Security processing by the recipient is mandatory** checkbox, and then click **Next**.
- The SOAP ACTION screen appears. Click **Next**.
- The TIME TO LIVE screen appears. Check the **Message expires** checkbox.
- Overwrite the **value** in the Time to live field (**120**) as the time to live after issued, and then click **Next**.

Note: The Time to live may have 1 to 20 numeric characters. The default Time to expire is 1 minute.

For designing real-time processing tasks, the Time to expire should reflect the smallest window of opportunity which allows SAML requests to pass through the product, as well as maintain the highest level of security.

For testing purposes, the Time to expire attribute should be increased, allowing time to complete testing and not allowing SAML assertions to expire.

- The SECURITY TOKEN TYPE screen appears. Select the **X.509 binary token** radio button, and then click **Next**.
- The X.509 TOKEN screen appears. There are two options for configuring the token, and both are explained, but this instruction uses the Static option.

Note: Selecting the Dynamic, based on protocol certificate radio button applies the X.509 Token, revealed earlier in the User Identity and Access Control task, to this WS-Security Header. Therefore, to select this option, the User Identity and Access Control task must have been added before the WS-Security Header task.

In later sections, review operations that use the Dynamic, based on protocol certificate option for WS-Security Headers and SAML Assertions.

- To configure the Token based on a specified user, click the **Static, based on a specified user** radio button.

Note: Selecting the Static, Based on Specified User radio button applies the X.509 Token of the certificate to this WS-Security Header.

- From the Specified certificate drop down list, click a **User Name**, and then click **Finish**.

Add WS-Security Header with SAML Assertion X.509 DN Token

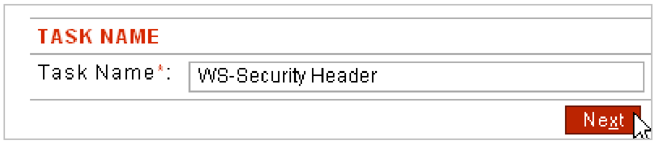
Follow these steps to add a WS-Security Header with a SAML Assertion X.509 Distinguished Name Token to a message that did not previously have one.

Use Case

This instruction uses SoapDocument.xml as the sample XML document loaded in the Task List. This file is available under the Samples directory of the supplied CD.

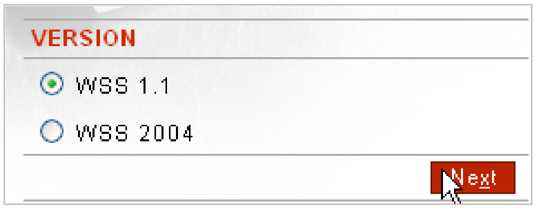
This operation adds a SAML Assertion X.509 Distinguished Name Token inside a WS-Security Header for Jack Kantos. The resulting document will be wrapped in a SOAP envelope.

Note: Before performing this operation, review the Prerequisites for All WS-Security Tasks listed earlier in this section.



TASK NAME

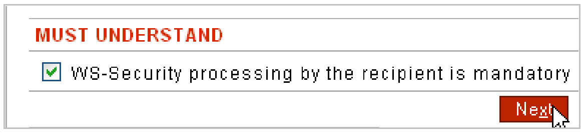
Task Name*: WS-Security Header



VERSION

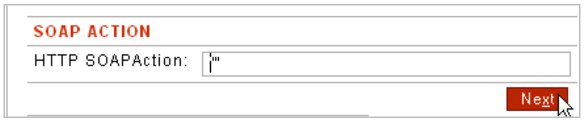
● WSS 1.1

○ WSS 2004



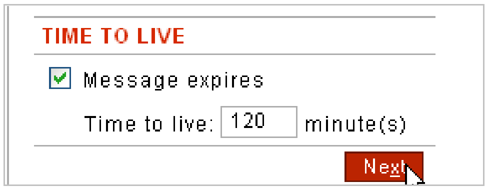
MUST UNDERSTAND

☑ WS-Security processing by the recipient is mandatory



SOAP ACTION

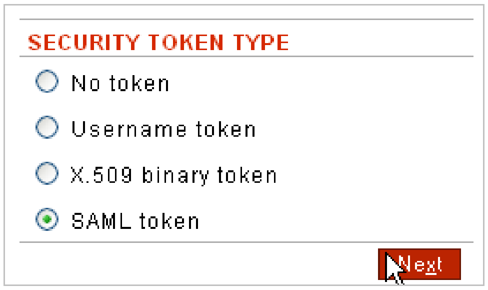
HTTP SOAPAction: ""



TIME TO LIVE

☑ Message expires

Time to live: 120 minute(s)



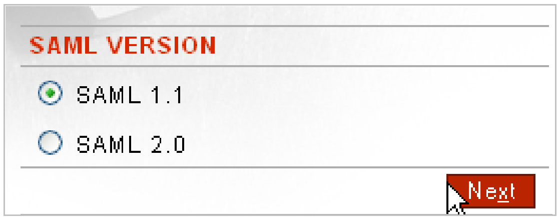
SECURITY TOKEN TYPE

○ No token

○ Username token

○ X.509 binary token

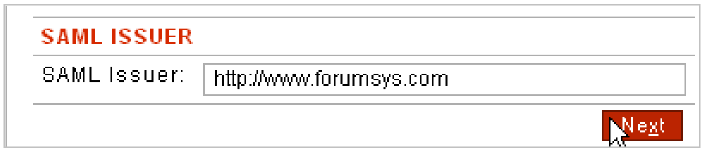
● SAML token



SAML VERSION

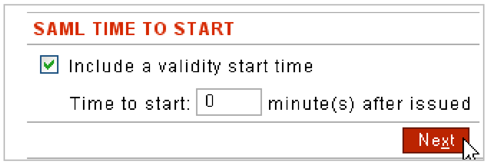
● SAML 1.1

○ SAML 2.0



SAML ISSUER

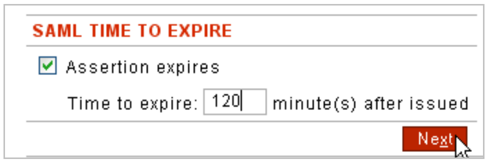
SAML Issuer: http://www.forumsys.com



SAML TIME TO START

☑ Include a validity start time

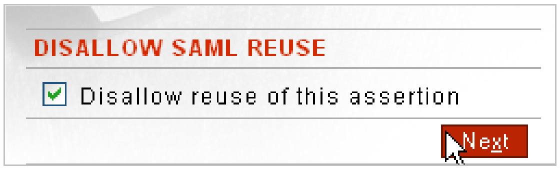
Time to start: 0 minute(s) after issued



SAML TIME TO EXPIRE

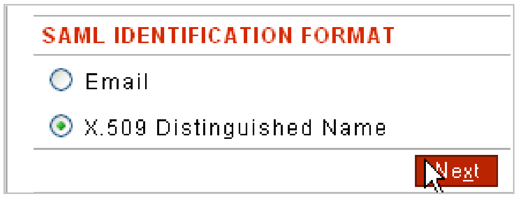
☑ Assertion expires

Time to expire: 120 minute(s) after issued



DISALLOW SAML REUSE

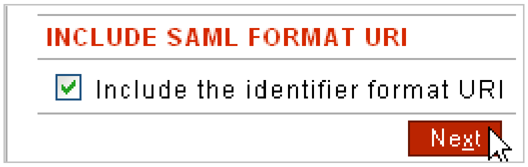
☑ Disallow reuse of this assertion



SAML IDENTIFICATION FORMAT

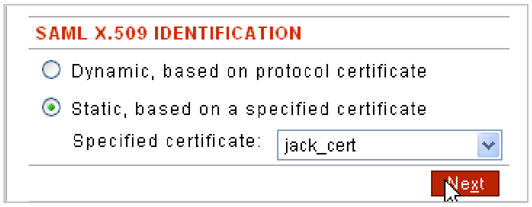
○ Email

● X.509 Distinguished Name



INCLUDE SAML FORMAT URI

☑ Include the identifier format URI

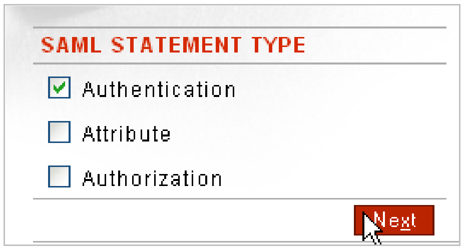


SAML X.509 IDENTIFICATION

○ Dynamic, based on protocol certificate

● Static, based on a specified certificate

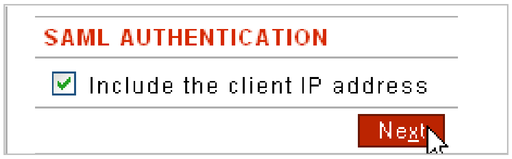
Specified certificate: jack_cert


SAML STATEMENT TYPE

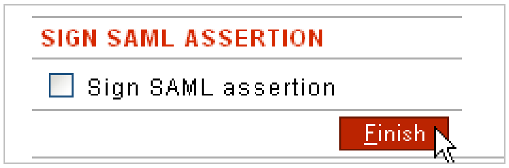
☑ Authentication

☐ Attribute

☐ Authorization


SAML AUTHENTICATION

☑ Include the client IP address


SIGN SAML ASSERTION

☐ Sign SAML assertion

- From the **TASK LIST** screen, from the Sample Document drop down list, select **SoapDopcument.xml**.
- Select **New** and the TASK TYPE screen appears.
- Select the **WS-Security Header** radio button, and then click **Next**.
- The TASK NAME screen appears. Click **Next**.
- On the VERSION screen, select the **WSS 1.1** or **WSS 2004** radio button, and then click **Next**.
- The MUST UNDERSTAND screen appears. Check the **WS-Security processing by the recipient is mandatory** checkbox, and then click **Next**.
- The SOAP ACTION screen appears. Click **Next**.
- The TIME TO LIVE screen appears. Check the **Message expires** checkbox.
- Overwrite the **value** in the Time to live field (**120**) as the time to live after issued, and then click **Next**.

Note: The Time to live may have 1 to 20 numeric characters. The default Time to expire is 1 minute.

For designing real-time processing tasks, the Time to expire should reflect the smallest window of opportunity which allows SAML requests to pass through the product, as well as maintain the highest level of security.

For testing purposes, the Time to expire attribute should be increased, allowing time to complete testing and not allowing SAML assertions to expire.

- The SECURITY TOKEN TYPE screen appears. Select the **SAML token** radio button, and then click **Next**.
- On the SAML VERSION screen, select the **SAML 1.1** or **SAML 2.0** radio button, and hten click **Next**.
- The SAML ISSUER screen appears. Accept the pre-populated SAML Issuer, and then click **Next**.
- The SAML TIME TO START screen appears. Check the **Include a validity start time** checkbox.
- Accept the time to start default value (0), and then click **Next.**
- The SAML TIME TO EXPIRE screen appears. Check the **Assertion expires** checkbox.
- Overwrite the **value** in the Time to expire field (**120**), and then click **Next**.

Note: The SAML Time to Expire header attribute may have 1 to 20 numeric characters. The default SAML Time to Expire is 1 minute.

For testing purposes, the SAML Time to Expire header attribute should be increased, allowing time to complete testing and not allowing SAML assertions to expire.

For designing real-time processing tasks, the SAML Time to Expire header attribute should reflect the smallest window of opportunity which allows SAML requests to pass through the product, as well as maintain the highest level of security.

- On the DISALLOW SAML REUSE screen, check the **Disallow reuse of this assertion** checkbox, and then click **Next**.
- The SAML IDENTIFICATION FORMAT screen appears. Select the **X.509 Distinguished Name** radio button, and then click **Next**. The INCLUDE SAML FORMAT URI screen appears.
- Check the **Include the identifier format URI** checkbox, and then click **Next**. The SAML X.509 IDENTIFICATION screen appears.
- There are two options for configuring the token, and both are explained, but this instruction uses the Static option.

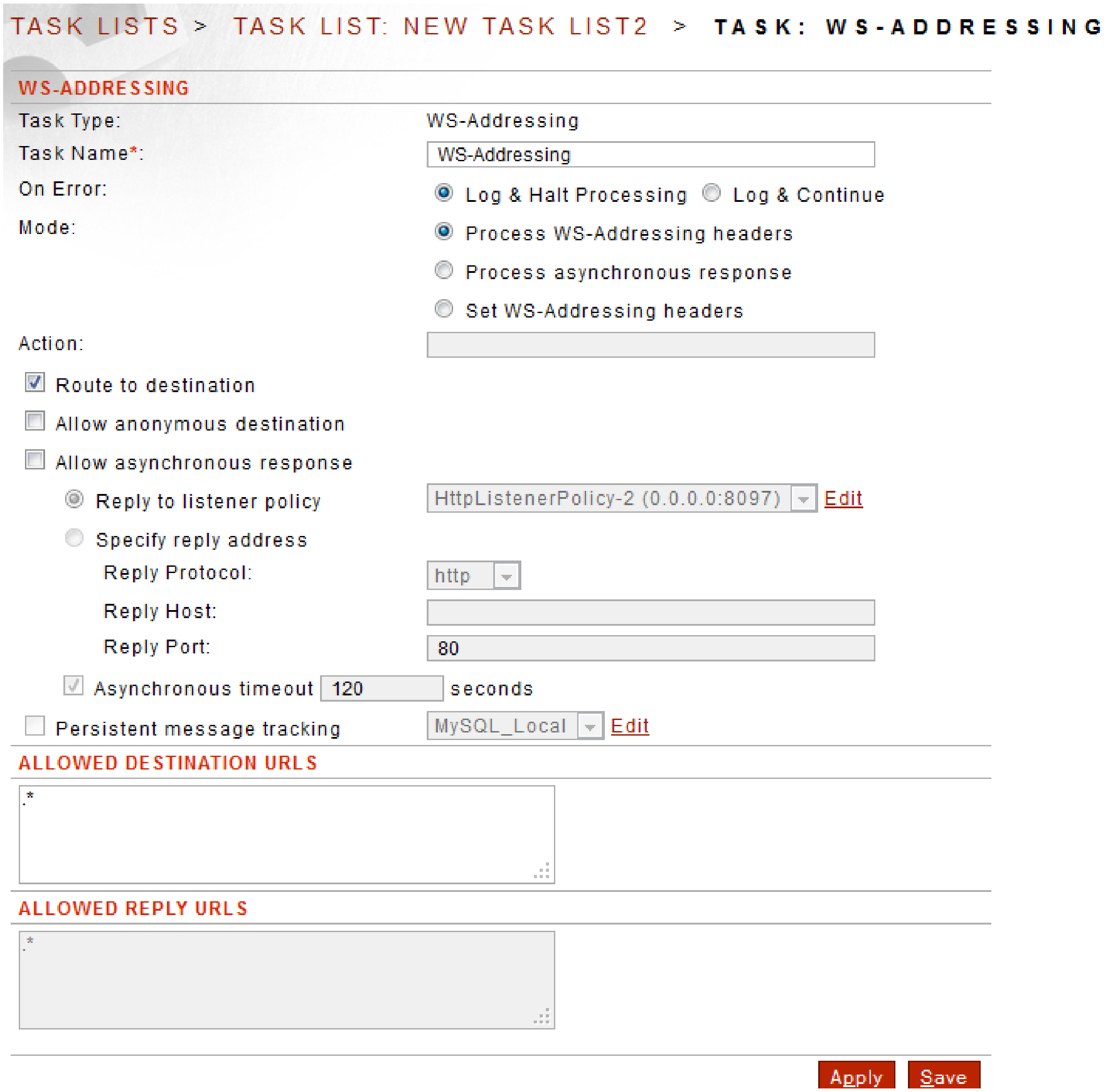
Note: Selecting the Dynamic, based on established identity radio button applies the X.509 Distinguished Name Token, revealed earlier in the User Identity and Access Control task, to this WS-Security Header. Therefore, to select this option, the User Identity and Access Control task must have been added before the WS-Security Header task.

Later sections of this document show the user of the Dynamic, based on Established Identity option for WS-Security Headers and SAML Assertions.

- To configure the Token based on a specified user, click the **Static, based on a specified user** radio button. From the User policy drop down list, click a **User Policy** name. Click **Next**. The SAML STATEMENT TYPE screen appears.
- Select the **Authentication** checkbox, and then click **Next**. The SAML AUTHENTICATION screen appears.
- Check the **Include the client IP address** radio button, and then click **Next**. The SIGN SAML ASSERTION screen appears.
- Skip the Sign SAML assertion option, and then click **Finish**.

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*: WS-Addressing

On Error: ● Log & Halt Processing ○ Log & Continue

Mode:
● Process WS-Addressing headers
○ Process asynchronous response
○ Set WS-Addressing headers

Action:

☑ Route to destination

☐ Allow anonymous destination

☐ Allow asynchronous response

● Reply to listener policy HttpListenerPolicy-2 (0.0.0.0:8097) Edit

○ Specify reply address

Reply Protocol: http

Reply Host:

Reply Port: 80

☑ Asynchronous timeout 120 seconds

☐ Persistent message tracking MySQL_Local Edit

ALLOWED DESTINATION URLS

```
.*
```

ALLOWED REPLY URLS

```
.*
```

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

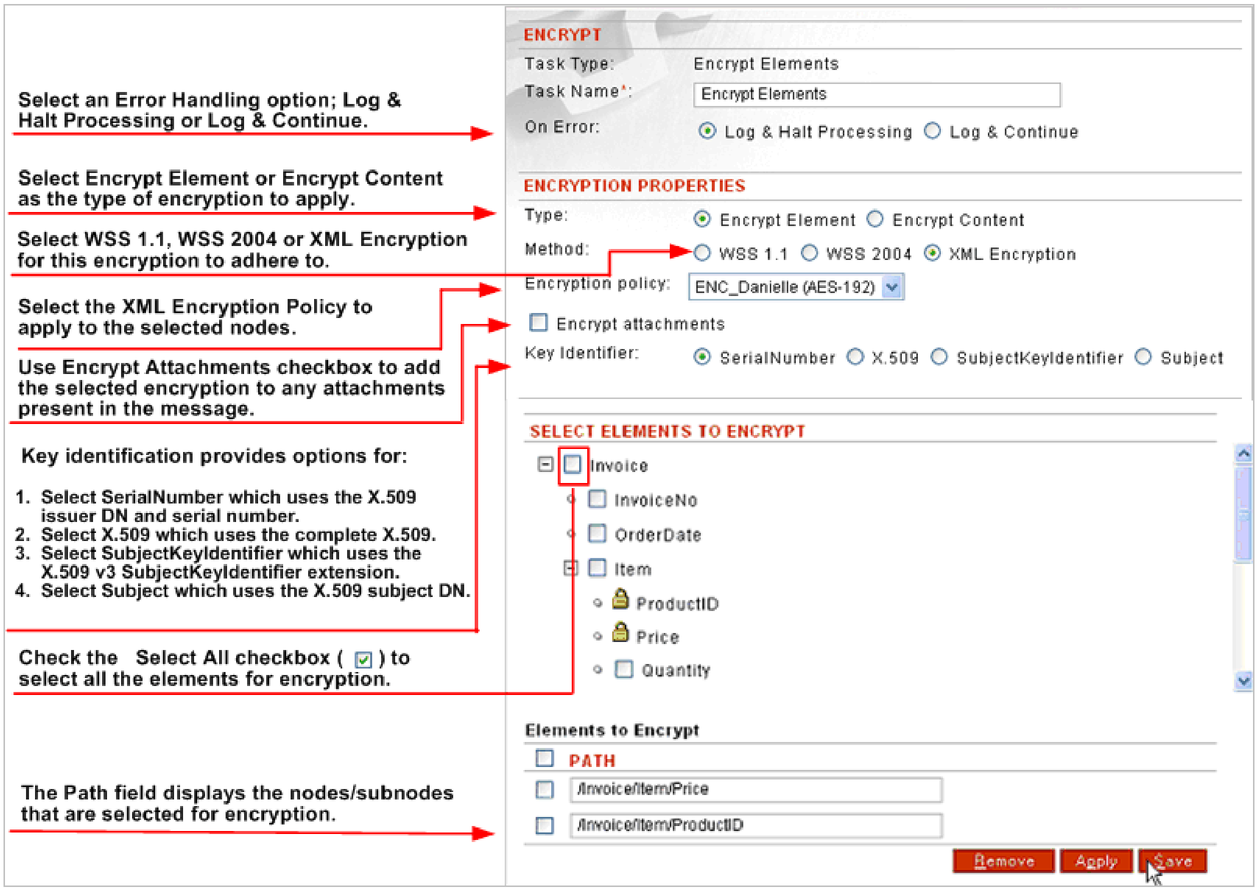
APPENDIX

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:



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.

ENCRYPT

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

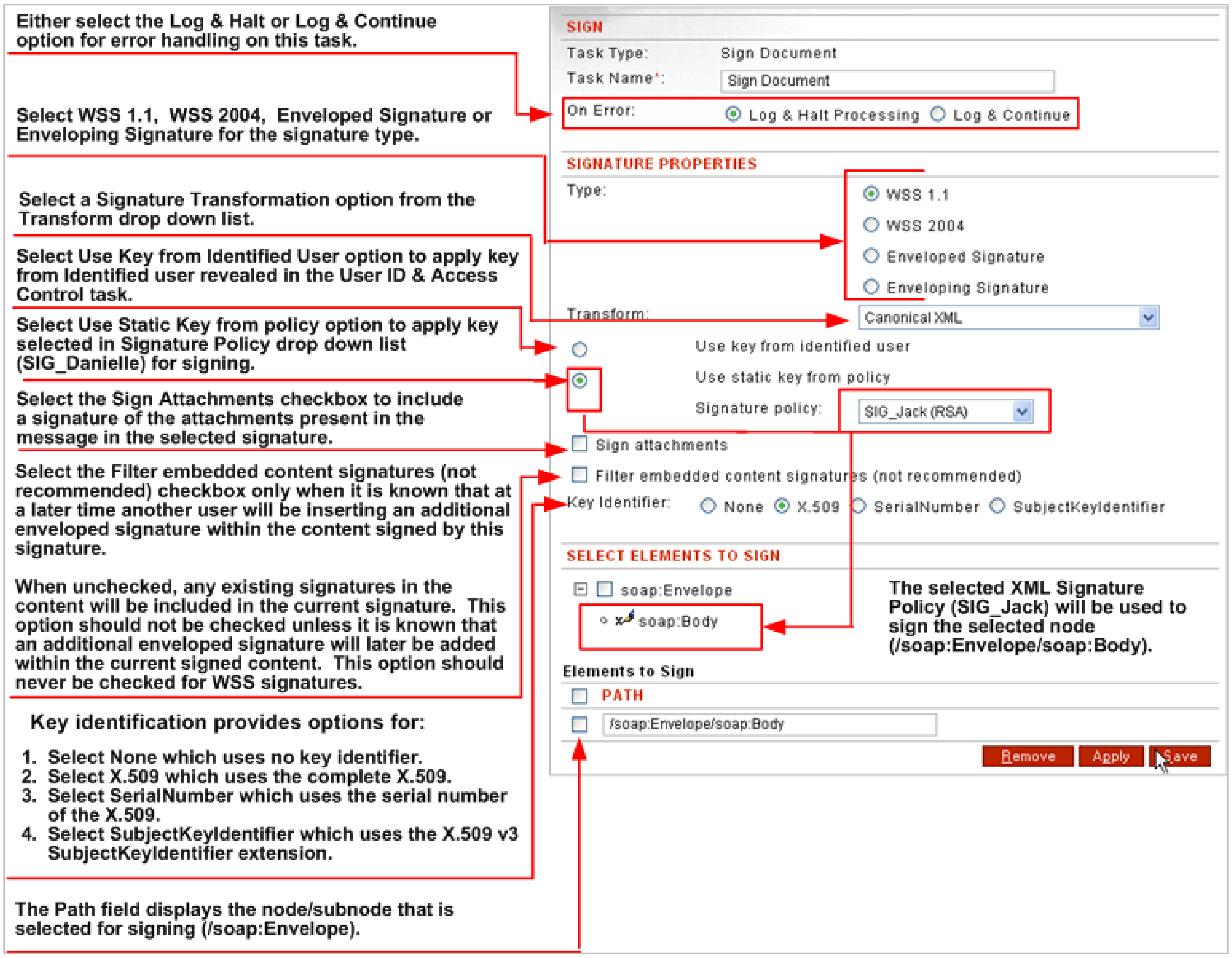
□ /Invoice/Item/Price

□ /Invoice/Item/ProductID

Figure 7: 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:



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).

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

Figure 8: 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.

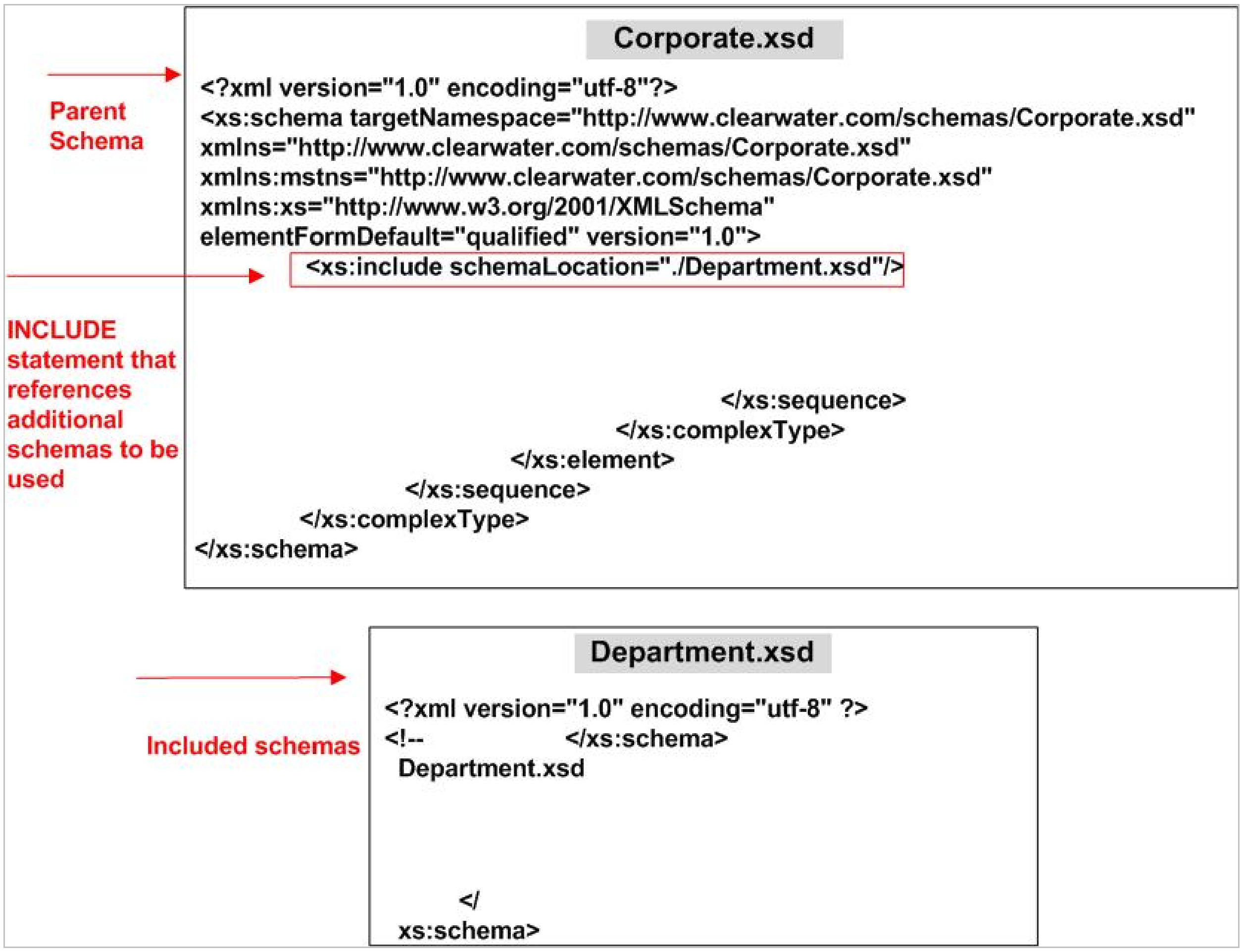
Appendix D - Example Compound Schema Reference Chart in Tasks Management Guide

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:



Parent Schema
INCLUDE statement that references additional schemas to be used
Included schemas

Corporate.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace="http://www.clearwater.com/schemas/Corporate.xsd"
xmlns="http://www.clearwater.com/schemas/Corporate.xsd"
xmlns:mstns="http://www.clearwater.com/schemas/Corporate.xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" version="1.0">
    <xs:include schemaLocation="./Department.xsd"/>



                              </xs:sequence>
                         </xs:complexType>
                    </xs:element>
               </xs:sequence>
          </xs:complexType>
</xs:schema>
```

Department.xsd

```
<?xml version="1.0" encoding="utf-8" ?>
<!--             </xs:schema>
 Department.xsd



     </
 xs:schema>
```

Figure 9: Example Compound Schema.

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

INDEX