



# **FORUM SENTRY™ VERSION 9 XML POLICIES GUIDE**



### **Legal Marks**

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

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

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

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

Forum Sentry™ Version 9 XML Policies Guide, published May 2024.

D-ASF-SE-490837

## Table of Contents

INTRODUCTION TO THE XML POLICIES GUIDE .....	4
Audience for the XML Policies Guide .....	4
Conventions Used in the XML Policies Guide .....	4
XML POLICIES .....	6
XML Features .....	7
XML Policy Examples .....	7
VIRTUAL DIRECTORIES .....	9
Virtual Directories Tab Screen Terms for XML Policies .....	10
Virtual Directory Detail Terms for XML Policies.....	10
Virtual Path Asterisk (*) Wildcard Limitations .....	12
Processing in Proxy and Service Modes .....	13
Protocol Mixing with XML Policies .....	14
Default Filter Expression in a Virtual Directory .....	15
TASK LISTS AND TASK LIST GROUPS FOR XML POLICIES.....	17
Task Lists Groups at the Virtual Directory Level .....	17
Task Lists Groups at the XML Policy Level .....	17
SETTINGS FOR XML POLICIES .....	19
IDP RULES FOR XML POLICIES .....	20
IDP Rule Tab Screen Terms for XML Policy .....	20
LOGGING SETTINGS FOR XML POLICIES.....	20
Logging Tab Screen Terms for XML Policy.....	20
TRANSFERRING EXPORTING AND IMPORTING XML POLICIES .....	21
Request Filters Available to All XML Policies .....	24
Request Filters Available to Each Virtual Directory .....	25
Content Types for Request Filters .....	28
Common Default Request Filters with XML Policies .....	30
Request Filter Syntax .....	30
View or Restore Common Default Request Filters for XML Documents .....	31
Delete a Request Filter .....	32
APPENDIX .....	34
Appendix A - How Request Filters Work .....	34
Appendix B - How to Invoke a Request.....	35
Appendix C - Sample Request Filters .....	37
Appendix D - Constraints in XML Policies Guide .....	38
Appendix E - Specifications in XML Policies Guide.....	38
Appendix F - Virtual Directory Reference Chart in XML Policies Guide.....	39
INDEX .....	40

## List of Figures

Figure 1: Proxy and Service Modes.....	13
Figure 2: Protocol Mixing on XML Policies. ....	14
Figure 3: Request Filters Identify and Convert XML Documents.....	34
Figure 4: The Virtual Directories Screen and Associated Options with XML Policies. ....	39

# INTRODUCTION TO THE XML POLICIES GUIDE

## Audience for the XML Policies Guide

The *Forum Systems Sentry™ Version 9 XML Policies Guide* for System Administrators who will:

- Create or import XML policies.
- Manage Virtual Directories on an XML policy.
- Manage settings on an XML policy.
- Associate IDP Groups to XML policies.
- Apply a Task List Group to an XML policy.
- Apply a Pattern Match policy to XML requests/responses.
- Apply Request Filter Templates on an XML policy.

## Assumptions

This document also assumes that the reader is familiar with the *Forum Systems Sentry™ Version 9 Web-based Administration Guide*.

For information on Task Lists and performing Tasks on an XML policy, refer to the *Forum Systems Sentry™ Version 9 Tasks Management Guide*.

## Screen Element on Legacy Systems

For customers upgrading from earlier versions of Forum Systems software to V9, the DOCUMENTS tab will not be visible on XML policies.

However, for customers running legacy versions, the DOCUMENTS tab will be visible. The Documents listed in the DOCUMENTS tab will not appear in the Documents screen (on the Navigator), but remain in the XML policies to be available for those customers who have run previous versions of the software.

## Conventions Used in the XML Policies 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: **\*\*\*\*\***

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

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

Throughout this and other documents in the Documentation Set, repetitive actions such as the following are not shown:

- View / edit a policy.
- Enable / disable a policy.
- Delete a policy.
- Rename a policy.
- Limit display of policies with Search or Max Results fields.

(For more information, refer to the Common Operations section of the *Forum Systems Sentry™ Version 9 Web-based Administration Guide*.)

Customers with plug-in licenses on the system, confirm that your 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.

For the focus of this document, the STATUS column is displayed on XML policies, and Virtual Directories.

### XML POLICIES

Search Usage: type any text  
No Labels

Filter Usage: type or select the label

<input type="checkbox"/>	NAME	VIRTUAL DIRECTORY	STATUS
<input type="checkbox"/>	<a href="#">Baltimore</a>	New Virtual Directory	<span style="color: green;">●</span>
<input type="checkbox"/>		New Virtual Directory2	<span style="color: green;">●</span>
<input type="checkbox"/>	<a href="#">Boston</a>	New Virtual Directory	<span style="color: green;">●</span>
<input type="checkbox"/>	<a href="#">Chicago</a>	New Virtual Directory	<span style="color: green;">●</span>
<input type="checkbox"/>	<a href="#">DenverOfficeXML</a>	New Virtual Directory	<span style="color: green;">●</span>
<input type="checkbox"/>	<a href="#">New XML Policy</a>	New Virtual Directory	<span style="color: green;">●</span>

Virtual Directories
Task Lists

<input type="checkbox"/>	VIRTUAL DIRECTORY	STATUS
<input type="checkbox"/>	<a href="#">New Virtual Directory</a>	<span style="color: green;">●</span>
<input type="checkbox"/>	<a href="#">New Virtual Directory2</a>	<span style="color: yellow;">●</span>

Request Filters, however, have a status of Enabled or Disabled only.

### REQUEST FILTER POLICY

Policy Name\*:
Default

<input type="checkbox"/>	#	MESSAGE TYPE FILTER	FORMAT	DESCRIPTION	STATUS
<input type="checkbox"/>	1	<a href="#">XML Default</a>	Simple	Plain XML	<span style="color: green;">●</span>
<input type="checkbox"/>	2	<a href="#">HTTP GET</a>	Simple	HTTP GET	<span style="color: green;">●</span>
<input type="checkbox"/>	3	<a href="#">Multipart</a>	Multipart	WSDL 1.1 MIME Filter	<span style="color: red;">●</span>
<input type="checkbox"/>	4	<a href="#">DIME</a>	DIME	WS-Attachments	<span style="color: red;">●</span>
<input type="checkbox"/>	5	<a href="#">Streaming</a>	Streaming	Generic	<span style="color: red;">●</span>
<input type="checkbox"/>	6	<a href="#">MTOM</a>	MTOM	SOAP Message Transmission Optimization Mechanism	<span style="color: red;">●</span>
<input type="checkbox"/>	7	<a href="#">SMTP Text</a>	Simple	HTML or Plain Text Email	<span style="color: red;">●</span>
<input type="checkbox"/>	8	<a href="#">SMTP MIME</a>	Multipart	Email with Attachments	<span style="color: red;">●</span>

Enable
Disable
New
Delete
Update
Save
Restore Defaults

## XML POLICIES

An XML policy is a set of rules that provide a policy for processing of XML flowing through the system.

XML policies include the following accessible properties and actions; each of which manages a portion of the XML policy and is detailed later:

XML POLICIES > XML POLICY

---

**XML POLICY**

Policy Name: NewXMLPolicy

---

**Virtual Directories** Task Lists Settings IDP Rules Logging

---

Virtual Directories > Virtual Directory: New Virtual Directory

---

**VIRTUAL DIRECTORY**

Name\*: New Virtual Directory

Description:

Virtual URI: http://192.168.1.144:80(/.\*)?

Remote URI: https://icmmdsit1.dstest.irsnet.gov:8443\$0

---

**VIRTUAL URI SETTINGS**

Listener Policy: ForumOAuthListener [Edit](#)

Virtual Host:

☐ Use virtual host as a regular expression

Virtual Path:

☐ Enable Virtual Path Case Insensitivity

Filter Expression: (/.\*)?

Replace Expression: \$0

Request Filter Policy: Default [Edit](#)

Error Template: [From Listener Policy]

---

**ACCESS CONTROL**

IP ACL Policy: Unrestricted [Edit](#)

ACL Policy: [Allow All]

XACML Policy: [None]

Password Authentication: [From Listener Policy]

Redirect Policy: [None]

---

**VIRTUAL DIRECTORY TASKS**

Request Task List Group: Task List Groups Type or select label --NONE--

Response Task List Group: Task List Groups Type or select label --NONE--

---

**REMOTE SETTINGS**

☒ Send to remote server

Remote Policy: FIR\_DSIT\_RemotePolicy [Edit](#)

Remote Path:

Host Header:

Process Response: On

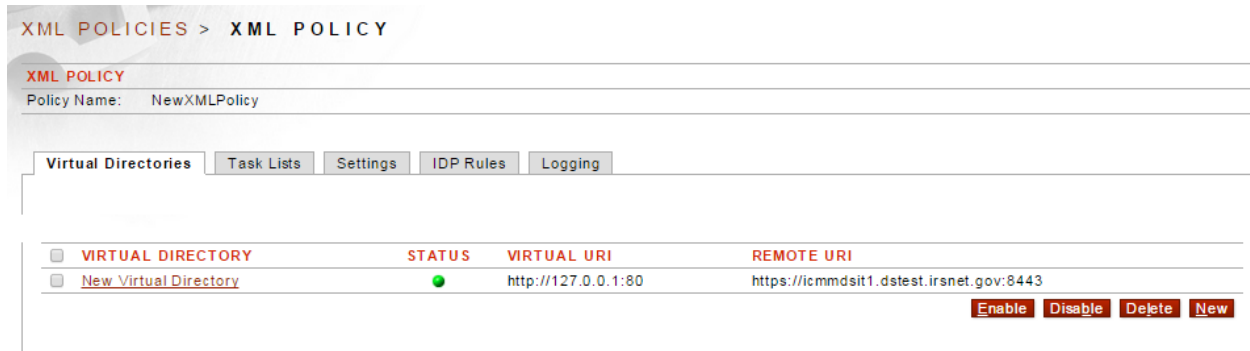
☐ Discard response from server

---

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

- **Virtual Directories:** Manage the services of the XML policy and Request Filters.
- **Task Lists:** Manage Task Lists Groups. (For more information on the Task Lists or performing Tasks, refer to the *Forum Systems Sentry™ Version 9 Tasks Management Guide*).
- **Settings:** Manage XML policy general settings.

- **IDP Rules:** Manage IDP Groups which represent a collection of Intrusion Detection and Prevention Rules. (For more information on IDP Rules, refer to the *Forum Systems Sentry™ Version 9 IDP Rules Guide*.)
- **Logging:** Manage policy level logging settings.



From an open XML policy, users may select:

- **Enable / Disable** to enable or disable the Virtual Directory.
- **Delete** to delete a Virtual Directory.
- **New** to create a new Virtual Directory.

## XML Features

An overview of the features available in a XML policy includes:

- Add an XML policy.
- Create new or associate existing listener and/or remote network policy.
- Add, view or edit virtual directories.
- Apply access control to virtual directories.
- Associate Task List Groups in the XML policy.
- Transfer, import or export XML policies. (For more information, refer to the *Forum Systems Sentry™ Version 9 System Management Guide*.)
- Edit the default HTTP Request Filter settings.

## XML Policy Examples

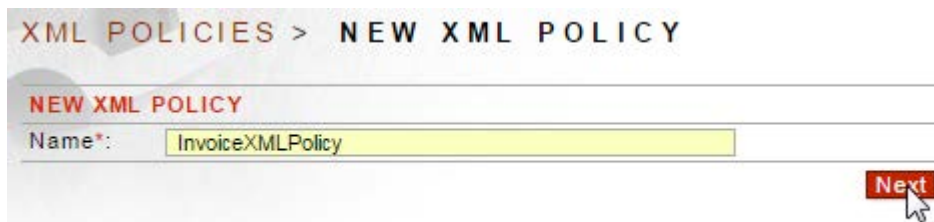
Examples for a XML policy include:

- Add an XML Policy.
- Create New Network Policies for XML Policy.
- Use Existing Network Policy for XML Policy.
- View Virtual Directories of an XML Policy.

### Add an XML Policy

When adding an XML policy, you may associate any existing Listener policy (HTTP, HTTPS, FTP, Tibco RV, Tibco EMS, IBM MQ, or Group Remote policy). Follow these steps to add an XML policy and associate an existing Listener policy:

## Adding an XML Policy



XML POLICIES > NEW XML POLICY

NEW XML POLICY

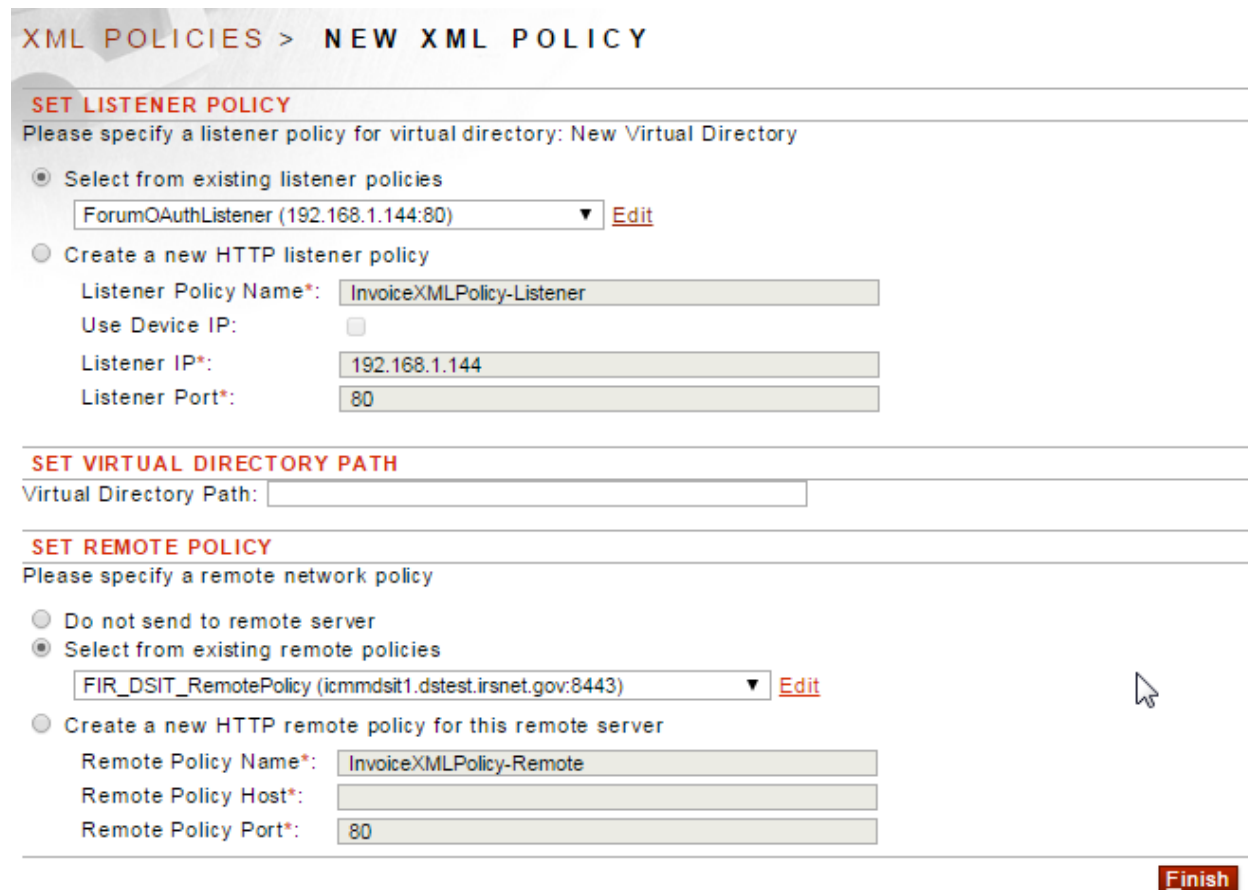
Name\*: InvoiceXMLPolicy

Next

- Navigate to the **XML Policies** screen and select **New**.
- In the Name field, enter the **Name** for this XML policy.
- In the Description field, enter a **Description** for this XML policy (optional), and then click **Next**. The SET LISTENER POLICY screen appears.

**Note:** At this point, you could associate any existing listener policy or create a new listener policy. This instruction uses the **Select from an existing listener policies** option.

Create New or Use Existing Network Policy for the XML Policy  
You may create a new Listener Policy when creating an XML Policy:



XML POLICIES > NEW XML POLICY

SET LISTENER POLICY

Please specify a listener policy for virtual directory: New Virtual Directory

☒ Select from existing listener policies

ForumOAuthListener (192.168.1.144:80) Edit

☐ Create a new HTTP listener policy

Listener Policy Name\*: InvoiceXMLPolicy-Listener

Use Device IP: ☐

Listener IP\*: 192.168.1.144

Listener Port\*: 80

SET VIRTUAL DIRECTORY PATH

Virtual Directory Path:

SET REMOTE POLICY

Please specify a remote network policy

☐ Do not send to remote server

☒ Select from existing remote policies

FIR\_DSIT\_RemotePolicy (icmmdsit1.dstest.irsnet.gov:8443) Edit

☐ Create a new HTTP remote policy for this remote server

Remote Policy Name\*: InvoiceXMLPolicy-Remote

Remote Policy Host\*:

Remote Policy Port\*: 80

Finish



- From the SET LISTENER POLICY section, select the **Create a new HTTP listener policy** radio button.

**NOTE:** CHECKING THE USE DEVICE IP CHECKBOX MEANS THAT THE IP FROM WHICH THIS LISTENER POLICY LISTENS WILL BE THE SAME AS THE SYSTEM'S DEVICE IP.

- Enter the **Listener IP** address in the Listener IP field or check the **Use Device IP** checkbox to use the assigned device IP of the system.
- Enter the **Listener Port** in the Listener Port field.
- Enter the **Virtual Directory URI** path for accessing this policy (here users can “cloak” the back-end URI by entering a value different from the actual physical URI of the back-end server).
- From the SET REMOTE POLICIES section, select the **Create a new HTTP remote policy for this remote server** radio button.

**Note:** The Virtual URI is a read-only field because the system determines this value from the Network policy, virtual path, Filter and Replace Expression settings. The Physical Path and Physical URI fields are read-only because the system uses the values from the XML document.

If Administrators need to allow arbitrary subdirectories or URL parameters, the Filter Expression can be changed from the default “/?” to “/.?\*” .

- Enter the **Remote IP** in the Remote policy Host field.
- Enter the **Remote Port** in the Remote policy Port field.
- Click **Finish**.

You may also use an existing Network Listener policy or Remote policy.

## VIRTUAL DIRECTORIES

The Virtual Directories tab displays a summary of all the Virtual URIs in this XML policy, as well as the Virtual URI and the Remote URI. XML policies can have multiple Virtual Directories, but each must either have a unique Virtual URI or specify a unique Virtual Host.

Clicking on the **Virtual Directory name** link reveals the Virtual Directory settings for this XML policy. Each virtual directory is used to map a virtual URI (local) to the physical path and URI (remote, as defined in the XML document).

**NOTE:** WHERE HTTP POLICIES ARE DISCUSSED, ALL OTHER NETWORK POLICIES ARE VALID, EXCEPT WHEN USING AN FTP POLICY AS A REMOTE POLICY.

A Virtual Directory is a pattern which matches an incoming HTTP request URI. A Virtual Directory is defined on the port node in an XML policy. Because the physical endpoint defined in the XML policy is static, virtual directories can be used to:

- Group different users according to their individual access control.
- Expose a different URI than the physical back end server URI.

## Virtual Directories Tab Screen Terms for XML Policies

The following table describes each term and definition on the Virtual Directories tab in XML policies.

TERM	DEFINITION
Virtual Directory	Local URIs used to access the XML policy.
Status	<ul style="list-style-type: none"><li>• Green status light = enabled policy.</li><li>• Yellow status light = a required functional element of this policy is disabled; i.e. the listener is disabled or the remote network policy is disabled.</li><li>• Red status light = disabled policy.</li></ul>
Virtual URI	The Unique Resource Indicator (URI) path used by clients to access this policy.
Remote URI	Actual URI back-end server.

## Virtual Directory Detail Terms for XML Policies

The following table describes each term and definition found on the Virtual Directory of an XML policy.

TERM	DEFINITION
Name	The identifier of this Virtual Directory.
Description	An optional description of this Virtual Directory.
Listener Policy	The Listener Policy on the system to associate with this Virtual Directory.
User Virtual Host as a Regular Expression	Using regular expressions within the virtual host definitions allow the HOST header to be matched based on the defined regular expression pattern. Enable this checkbox if the value entered in the virtual host field is to be interpreted as a regular expression rather than a string match for comparing to the inbound HOST header.
Virtual Host	<p>The Virtual Host option allows the IP:Port combination to have a 3rd parameter which uses the HOST header of the inbound request to determine which virtual directory policy matches. With no virtual host defined, the virtual directory is matched simply based on IP, Port and URI. With virtual host defined, the virtual directory is matched based on IP, Port, HOST Header, and URI.</p> <p>i.e.</p> <p><a href="http://10.5.1.1:80/test/policy">http://10.5.1.1:80/test/policy</a> HOST: prod.company.com</p> <p><a href="http://10.5.1.1:80/test/policy">http://10.5.1.1:80/test/policy</a> HOST: dev.company.com</p>
Virtual Path	The Virtual Path field allows users to customize this XML's virtual path.
Virtual URI	The Unique Resource Indicator (URI) path used by clients to access this policy. This is where the system receives a request.

Filter Expression	The default "/" value represents an extended regular expression on which exists a trailing portion that must match a defined pattern before a request is accepted for processing.
Replace Expression	The "\$0" value represents the entire trailing portion of the request URI.
Send to remote server	<ul style="list-style-type: none"> <li>When checked, the Remote Policies drop down list is enabled. Now, all requests and responses will be processed by the system in Proxy mode and sent to the selected Remote Policy.</li> <li>When unchecked, all requests and responses will be processed by the system in Service mode, with the processed request being returned to the client, and access to the Remote policy is disabled.</li> </ul> <p>For more on Proxy versus Service mode see the chapter below titled: Processing in Proxy and Service Modes</p>
Discard response from server	When checked, responses from the back-end server are discarded.
Remote Policy	The Remote Policy associated with this Virtual Directory.
Remote Path	The back-end server IP / Port which identifies the Remote Policy.
Remote URI	Actual URI back-end server.
Host Header	The Host header set by Sentry when communicating with the remote server.
Process Response	When set to ON, the response from the back-end server undergoes pre-processing before being sent to the client.
IP ACL	The IP Access Control List that will be enforced on this Virtual Directory. With Unrestricted selected, there is no access control by IP enforced.
ACL	The User Access Control List that will be enforced on this Virtual Directory. With the Allow All ACL selected, there is no access control enforced. The selected User ACL grants access of this XML policy to any member of the User ACL.

Password Authentication	<p>When set to From Listener Policy, the password authentication credentials captured at the Listener Policy level will be used for enforcement.</p> <p>When set to Specify, the administrator can choose to enforce any of the following Password Authentication options:</p> <ul style="list-style-type: none"> <li>• Use basic authentication</li> <li>• Use digest authentication</li> <li>• Use cookie authentication</li> <li>• Use form post authentication</li> <li>• Username and Password Parameters are used with the form post authentication</li> <li>• Require password authentication (any): to enforce a successful authentication not just capture the credentials.</li> </ul> <p>For more information on Password Authentication please refer to the Forum Sentry v9 Access Control Guide.</p>
Redirect Policy	The Redirect Policy that is associated to this Virtual Directory. Redirect Policies allow redirection to a different URL based on four events: Authentication Success, Authentication Failure, No Credentials and On Error. A valid Redirect Policy will need to be configured on the Resources>>Redirect Policies page in order to associate a Redirect Policy to the Virtual Directory.
Error Template	Associate an Error Template to this Virtual Directory or reference the Error Template in a selected Listener Policy that is associated with this Virtual Directory.
Request Task List Group	The Task List Group selected to process the request messages for this Virtual Directory.
Response Task List Group	The Task List Group selected to process the response message for this Virtual Directory.

For information on HTTP Request Filters, refer to the Request Filters for XML Policies section of this document.

### Virtual Path Asterisk (\*) Wildcard Limitations

Consecutive asterisk wildcards are not supported. For example, 'project/facility/\*/88/room' is equivalent to 'project/facility/\*/88/room'. The table below illustrates valid matches given a pattern.

PATTERN	VALID MATCHES
project/facility/*/wing*/room	project/facility/1/wing/2/room project/facility/3/wing/4/room project/facility/theawesomefacility/wing/thecoolwing/room
project/facility/*/88/room	project/facility/1/88/room project/facility/2/88/room project/theawesomefacility/88/room

## Operations on Virtual Directories for XML Policies

XML policies may have one or more Virtual Directories. Operations on Virtual Directories include:

- Add, edit or associate another Listener and/or Remote policy to the Virtual Directory.
- Configure Additional Virtual Directories on an XML policy.
- View / reconfigure a Virtual Directory.
- Enable / disable the Virtual Directory.
- Associate an ACL policy to the Virtual Directory.
- Associate an Error Template to this Virtual Directory or reference the Error Template in the Listener Policy.
- Edit the Remote Path of this Virtual Directory.
- Edit the Filter Expression used.
- Change the Replace Expression used.
- Add, edit, enable/disable, remove, promote or demote the request filter associated with the Virtual Directory.
- Select a Redirect Policy for the Virtual Directory.

Virtual Directories in XML policies may be set to process traffic in proxy mode or service mode.

## Processing in Proxy and Service Modes

The following graphic displays processing in Proxy or Service modes:



Figure 1: Proxy and Service Modes.

## Proxy Mode

In Proxy mode, a document is sent from the client to the appliance, processed, sent to the back end server, processed, returned to the appliance for optional processing, and then returned to the client. Proxy mode is set when:

- the **Sent to Remote Server** checkbox in the Virtual Directory is checked.
- a **Remote policy name** is selected in the Remote policy field in the Virtual Directory.

## Service Mode

Service mode allows the product to run as a service provider. A client request is processed by the product as an XML document and then sent back to the client in the HTTP response. Service mode is set when:

- the **Sent to Remote Server** checkbox in the Virtual Directory is unchecked.
- access to the Remote policy field is blocked in the Virtual Directory.

## Protocol Mixing with XML Policies

Protocol mixing with XML policies provides a method of mixing protocols between incoming request and outgoing responses on the system. Protocol mixing is allowed on the following example network policies, from Incoming Request to Outgoing Response on the system:

- from HTTP/S listener to Tibco-Rv remote.
- from HTTP/S listener to Tibco-EMS remote.
- from HTTP/S listener to IBM MQ remote.
- from HTTP/S listener to SMTP remote.
- from Tibco-Rv listener to HTTP/S remote.
- from Tibco-EMS listener to HTTP/S remote.
- from IBM MQ listener to HTTP/S remote.
- from SMTP listener to HTTP/S remote.

**NOTE:** THE BULLETED LIST ABOVE DOES NOT CONTAIN **EVERY** PERMUTATION POSSIBLE WITH PROTOCOL MIXING, BUT IS A SMALL REPRESENTATIVE SUMMARY OF SOME PROTOCOLS THAT MAY BE MIXED WITH OTHERS.

## How the System Manages Protocol Mixing on XML Policies

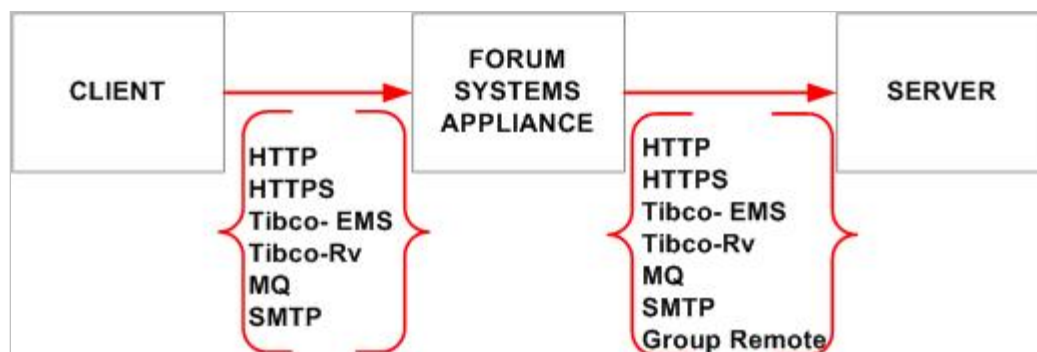


Figure 2: Protocol Mixing on XML Policies.

**NOTE:** FOR MORE INFORMATION, REFER TO THE MIX PROTOCOLS ON AN XML POLICY INSTRUCTION.

## Asynchronous Protocols Supported with XML Policies

The system also supports protocol mixing between the following asynchronous protocols:

- from Tibco-EMS to Tibco-Rv.
- from Tibco-EMS to IBM MQ.
- from Tibco-RV to Tibco-EMS.
- from Tibco-Rv to IBM MQ.
- from IBM MQ to Tibco-EMS.
- from IBM MQ to Tibco-Rv.

Asynchronous protocols, such as IBM MQ, need to be used in the “synchronous” mode in order to be compatible with HTTP. For example, if an IBM MQ policy has the Synchronous policy option turned off, protocol matching cannot occur with HTTP because they are incompatible paradigms.

### Authentication with IBM MQ Policies

When authenticating a message in an IBM MQ policy or Tibco-EMS policy during run-time, the system searches each message for the **fs\_user** and **fs\_password** property, and uses this information to authenticate each message and establish identity.

For the JMS-based messaging protocols that support SSL (Tibco EMS, IBM MQ) we have added our own basic authentication capability to allow each message to be authenticated and an identity established. The identity can then be used for access control, obtaining a signing key or even generating and propagating an identity token such as a SAML token. The sender simply has to add two fields to the message headers that contain the user and password to use. For protocols that support SSL, it is recommended that SSL is used when sending the password along with a message. The password will not be propagated after it is consumed by the system. The properties **fs\_user** and **fs\_password** should be used in the JMS headers to add the appropriate credentials.

### HTTP Headers

When HTTP is the inbound protocol, all headers allowed by RFC 2616 may be propagated to the remote protocol. The converse is also true, if the listener protocol is a JMS protocol (Tibco EMS or IBM MQ) any http headers that are specified (escaped with underscores rather than dashes) and the remote protocol is HTTP the headers will be placed into the HTTP protocol and propagated. This allows cookies such as authentication tokens from Tivoli Access Manager to be propagated and also content-type and any other stateful headers to be passed.

When mixing protocols on an IBM MQ policy, for example, the system manages authentication by converting all dashes to underscores in HTTP headers. This allows for the case of | HTTP | ----- | IBM MQ | ----- |HTTP| and all of the inbound headers (and cookies) will be propagated.

### Default Filter Expression in a Virtual Directory

When a client request is received on a Virtual Directory at run time, the path of the client request URI consists of the Virtual Path followed by a trailing portion. The Filter Expression is an extended regular expression which the trailing portion must match before the request is accepted for processing.

To review the syntax of the Filter Expression follows Java's regular expression rules; refer to documentation at

<http://java.sun.com/j2se/1.5.0/docs/api/java/util/regex/Pattern.html>.

**NOTE:** THE DEFAULT FILTER EXPRESSION "/" IS MORE RESTRICTIVE THAN IN SOME PREVIOUS VERSIONS OF THE PRODUCT. IF YOU NEED TO ALLOW SUBDIRECTORIES OR URI PARAMETERS (A QUERY STRING), YOU CAN CHANGE THE FILTER EXPRESSION TO THE ALL-INCLUSIVE ".\*".

## Replace Expression in a Virtual Directory

When a client request starts with the virtual path and the trailing portion matches the Filter Expression, the trailing portion is replaced by the Replace Expression and appended to the physical URI (WSDL policies) or Remote URI (XML policies) when connecting to the remote server. In the Replace Expression, \$0 represents the entire trailing portion of the request URI. \$1 represents the portion of the request URL matched by the first set of parentheses in the Filter Expression (first capture group), \$2 represents the portion matched by the second set of parentheses, up through \$9. See the example below.

The default Replace Expression '\$0' means that the system will preserve the trailing portion of the client request URI in the remote request URI. The Replace Expression can be left empty to indicate that the Remote URI should not include the trailing portion at all.

Client requests are mapped to a Virtual Directory at run-time as follows:

1. The path of the client request URI is compared with the virtual path of each enabled Virtual Directory configured for the Listener policy the request was received on.
2. If more than one Virtual Directory matches, the most specific match is selected. For example, if Virtual Directories '/one' and '/one/two' are configured, a request for '/one/two/three' will be processed by the Virtual Directory with path '/one/two', while a request for '/one/four' will be processed by the Virtual Directory with path '/one'. If the Virtual Directory with path '/one/two' is subsequently disabled, both requests will now be processed by the Virtual Directory with path '/one'.
3. If no Virtual Directories match the request URI, the request is rejected with an error message stating that the requested Virtual Directory is not found.
4. Once a Virtual Directory is selected, the trailing portion of the request URI is matched against the Filter Expression. If the match fails, the request is rejected with an error message stating that the path match has failed. Other, less-specific Virtual Directories found in step 2 are **not** used in this case.

Example:

WSDL port Virtual Directory is configured with:

```
[ HTTP Listener policy IP: 10.1.0.1, port: 80 ]
Virtual Path: /virtual/service
Filter Expression: \?id=(u[0-9]{2})&food=([a-z]+)
Replace Expression: /fruit/$2;user=$1
[ Remote Path from WSDL: /remote ]
[ Physical URI: http://10.0.0.3/remote/fruit/$2;user=$1 ]
```

A client request comes in for the URL <http://10.1.0.1/virtual/service?id=u21&food=apple>.

The trailing portion is '?id=u21&food=apple' which matches the Filter Expression. In the Filter Expression, the first capturing group is '(u[0-9]{2})' which matches 'u21' from the request URL, and the second capturing group is '([a-z]+)' which matches 'apple' from the request URL.

Therefore, the request is proxied to a remote server using the following Physical URI:  
<http://10.0.0.3/remote/fruit/apple;user=u21>.



## TASK LISTS AND TASK LIST GROUPS FOR XML POLICIES

The Task List tab allows users to view all Tasks and Task Lists associated with an XML policy through Task List Groups.

**Note:** With Forum Systems Sentry v9, Task List Groups can now be set to process request or response documents individually per Virtual Directory, or per XML Policy.

### Task Lists Groups at the Virtual Directory Level

Task List Groups set at the Virtual Directory level are applicable only the Request or Response Messages for that Virtual Directory. Different Task List Groups can be selected for the request or response messages.

The screenshot displays the 'Virtual Directories' configuration page, specifically the 'Task Lists' tab for a 'New Virtual Directory'. The interface is organized into several sections:

- VIRTUAL DIRECTORY:** Fields for Name (New Virtual Directory), Description, Virtual URI (http://10.10.20.10:8181/testtesttest/), and Remote URI (http://api.openweathermap.org/testtesttest\$0).
- OPENAPI SETTINGS:** A checkbox for 'Publish a different location in exported OpenAPI' and fields for Published Protocol (http), Published Host, and Published Port.
- VIRTUAL URI SETTINGS:** Fields for Listener Policy (DEX-8181), Virtual Host, Virtual Path (/testtest), Filter Expression (/.\*?), Replace Expression (\$0), Request Filter Policy (Default\_HTML), Error Template ([From Listener Policy]), and Google Analytics ([None]).
- ACCESS CONTROL:** Fields for IP ACL Policy (Unrestricted), Host ACL Policy ([None]), ACL Policy ([Allow All]), Password Authentication ([From Listener Policy]), and Redirect Policy ([None]).
- VIRTUAL DIRECTORY TASKS:** Fields for Request Processing and Response Processing, both set to 'Task List Groups' with a dropdown for 'Type or select label name' and a '[None]' button.
- REMOTE SETTINGS:** A checkbox for 'Send to remote server' and fields for Remote Policy (api.openweathermap.org), Remote Path (/testtesttest), Host Header, and Process Response (On).

At the bottom right, there are 'Apply' and 'Save' buttons.

### Task Lists Groups at the XML Policy Level

Task List Groups set at the XML Policy level are applicable for all Virtual Directories of the XML Policy. The Task List Groups can be associated with the Request or Response Messages for all Virtual Directories. Different Task List Groups can be selected for the request or response messages.

## XML POLICIES > XML POLICY

### XML POLICY

Policy Name: NewXMLPolicy

Virtual Directories

**Task Lists**

Settings

IDP Rules

Logging

#### TASK LIST GROUPS

Request Task List Group

Task List Groups ▼ Type or select label ▼ --NONE-- ▼

Response Task List Group

Task List Groups ▼ Type or select label ▼ --NONE-- ▼

Create

Save

**NOTE:** FOR FULL DOCUMENTATION ON TASKS, TASK LISTS AND TASK LIST GROUPS, REFER TO THE *FORUM SYSTEMS SENTRY™ VERSION 9 TASKS MANAGEMENT GUIDE*.  
FOR INFORMATION ON EDITING / VIEWING A TASK LIST, REFER TO THE COMMON OPERATIONS OF THE *FORUM SYSTEMS SENTRY™ VERSION 9 WEB-BASED ADMINISTRATION GUIDE*.

## SETTINGS FOR XML POLICIES

The Settings tab includes name and description for this XML policy. The Settings tab also includes the “Protect virtual resource option” and the “Enable session cookies option.”

TERM	DEFINITION
Policy Name	The identifier of this XML Policy.
Policy Description	An optional description of this XML Policy.
Protect Virtual Resource	<p>When Protect virtual resource is checked, the system uses the Sentry Virtual Directory when authenticating and authorizing against the back end server.</p> <p>When Protect virtual resource is unchecked, the system uses the Sentry physical URI when authenticating and authorizing against the back end server.</p>
Enable Session Cookies	<p>When the Enable session cookies option is checked, Sentry will automatically set a cookie (often the FSESSION cookie) for authentication and cache it for the duration noted. The cookie can be used in a Single Sign On paradigm.</p> <p>When the Enable session cookies option is unchecked, cookie is set.</p> <p>Cookie Parameters include:</p> <ul style="list-style-type: none"> <li>• Cookie Name</li> <li>• Cookie Path</li> <li>• Cookie Domain</li> <li>• Session Timeout (mins)</li> <li>• Session Idle Timeout (mins)</li> </ul>
Enable Persistent Sessions	When the Enable Persistent Sessions option is checked, Sentry will store the cookie information in a database, using the selected Data Source. This allows for persistent sessions across multiple Sentry instances that all use the same database.
Use Secure cookies	A cookie with the Secure attribute is sent to the server only with an encrypted request over the HTTPS protocol, never with unsecured HTTP, and therefore can't easily be accessed by a man-in-the-middle attacker.
Use HTTP Only cookies	HttpOnly is an additional flag included in a Set-Cookie HTTP response header. Using the HttpOnly flag when generating a cookie helps mitigate the risk of client side script accessing the protected cookie (if the browser supports it)
WAF Policy	Associate a Web Application Firewall (WAF) policy from <b>Resources-&gt;WAF Policies</b>
Exclude from Monitoring	Do not include statistics from this policy in the Monitoring and performance statistics
Enable Response Caching	Enable a response caching policy (when licensed for this feature) to apply to responses for this policy
Enable Google Analytics	Enable statistics from this policy to be written to a Google Analytics policy (when licensed for this feature)
Enable Persistent Sessions	When the Enable Persistent Sessions option is checked, Sentry will store the cookie information in a database, using the selected Data Source. This allows for persistent sessions across multiple Sentry instances that all use the same database.

## IDP RULES FOR XML POLICIES

Intrusion Detection and Prevention (IDP) Rules define a set of criteria which can be associated with an XML policy. IDP Groups represent a reusable collection of IDP Rules that may be applied to this XML policy. Under the IDP Group drop down list is a listing of all the IDP Rules included in the selected IDP Group.

**NOTE:** FOR FULL DOCUMENTATION THAT THE PRODUCT PROVIDES ON IDP RULES, REFER TO THE *FORUM SYSTEMS SENTRY™ VERSION 9 IDP RULES GUIDE*.

IDP Rules also allow throttling and black listing based on identity, IP and traffic load. IDP Rules can be scheduled based on expected traffic to throttle back transactions or reroute messages.

IDP Rules have actions associated with them that can generate an email alert or invoke a specified web service, triggering any event programmed into the web service.

IDP Rules define a set of identified criteria used by the system to detect intrusion. Once created, IDP Rules may be reused.

### IDP Rule Tab Screen Terms for XML Policy

The following table describes each term and definition found on the IDP Rule tab.

TERM	DEFINITION
IDP Group	The identifier for this IDP Group.
IDP Rule	IDP Rules that is included in this IDP Group.
IDP Criterion	Description of the type of IDP Rule.
Threshold	Any constrained value, period or rate applied to the detection settings of the IDP Rule.
User Group	The name of the User group for which the IDP Rule applies.
Enforce By	<ul style="list-style-type: none"><li>• If User, the IDP Rule is enforced on a per User basis. If IP, the IP address that is defined in the detection settings of the IDP Rule.</li><li>• If IP, the IDP Rule is enforced on a per IP address User basis.</li></ul>
IDP Action	The name of the IDP Action policy applied to the IDP Rule.
IDP Schedule	The name of the IDP Schedule policy applied to the IDP Action.

## LOGGING SETTINGS FOR XML POLICIES

Policy level logging can be set for each XML Policy. This allows for logging different policies with different log levels.

### Logging Tab Screen Terms for XML Policy

The following table describes each term and definition found on the IDP Rule tab.

TERM	DEFINITION
Enable Policy Level Logging Settings	When checked, policy level logging is enabled for the XML Policy. When not checked, policy level logging is disabled for the XML Policy.
Policy Log Level	When policy level logging is enabled, this is the log level set for this policy.
Always Log the Following Code	When policy level logging is enabled, this is a list of error codes that will always be logged regardless of the log level set for this policy.
Pattern Match Policy	When policy level logging is enabled, and the Always log the following codes option is enabled, a pattern match policy can be used to log messages based on a pattern match policy (regex).

**Note:** For more information on logging with Sentry, please see the Forum Sentry v9 Logging Guide. For more information on Pattern Match policies, see the Forum Sentry v9 IDP Rules Guide.

## TRANSFERRING EXPORTING AND IMPORTING XML POLICIES

Users may transfer one or more XML policies (and all its dependencies) from one Agent machine to another Agent machine with the **GDM Transfer** command visible on the XML Policies screen. This type of transfer is referred to as a GDM partial configuration transfer.

Users may export one or more XML policies (and all its dependencies) to a local file system via an FSG file using the **GDM Export** command visible on the XML Policies screen. This type of export is referred to as a GDM partial configuration export.

Through the Import / Export screen, users may import XML policies with all their dependencies into the product using the **Import** command from the **GDM IMPORT** section of the screen. This type of import is referred to as a GDM partial configuration import.

For information on the following features, refer to the following sections of these volumes:

- To transfer an XML policy to an Agent Group, refer to the GDM Partial Configuration Transfer section of the *Forum Systems Sentry™ Version 9 System Management Guide*.
- To export an XML policy, to a local file system via an FSG file, refer to the GDM Partial Configuration Export section of the *Forum Systems Sentry™ Version 9 System Management Guide*.

**XML POLICIES**

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

**No Labels**

	NAME	VIRTUAL DIRECTORY	STATUS	VIRTUAL URI	REMOTE URI
<input type="checkbox"/>	<a href="#">New XML Policy</a>	<a href="#">New Virtual Directory</a>	<span style="color: green;">●</span>	<a href="http://127.0.0.1:80/n">http://127.0.0.1:80/n</a>	<a href="http://www.forumsys.com:80/n">http://www.forumsys.com:80/n</a>
<input type="checkbox"/>	<a href="#">NewXMLPolicy</a>	<a href="#">New Virtual Directory</a>	<span style="color: green;">●</span>	<a href="http://127.0.0.1:80">http://127.0.0.1:80</a>	<a href="http://www.forumsys.com:80">http://www.forumsys.com:80</a>

**GDM Transfer   GDM Export   Delete   New   Copy**

- To Import an XML policy with all its dependencies to the current machine via an FSG file, refer to the GDM Partial Configuration Import section of the *Forum Systems Sentry™ Version 9 System Management Guide*.

---

**GDM IMPORT**

---

Password\*:

☒ From file (.fsg)\*:

Choose File

No file chosen

☐ From database

Configuration Name:

Browse

New domain:

Do not change ▾

Import

## REQUEST FILTERS FOR XML POLICIES

A Request filter allows the system to select those HTTP requests that match selection criteria based on the HTTP headers and decode the request appropriately. Most request filters will only need to examine the content-type header, but any header may be used.

Request filters can be used to manage sets of standard, emerging and future content types, along with associated rules. Administrators may add, configure, edit and remove request filters, as well as restore default request filters that have been deleted. You may enable or disable request filters, and re-prioritize the list of request filters. Request filters include a name, format, description, identifying expression and parameter.

There are two sets of default Request Filters. One is pre-configured; the other one is not.

One set of Request Filters is common; that is, these are a collection of Request Filters which are available to all XML policies.

The other set of Request Filters is local; that is, these are a collection of Request Filters which are available to any subsequently created Virtual Directory on an individual XML policy.

Both sets of Request Filters include:

- XML Default\*
- Web Form\*
- Web Form Data
- HTTP GET\*
- Multipart
- DIME (Direct Internet Message Encapsulation)
- Streaming
- REST
- MTOM
- JSON

\* These Request Filters are enabled by default; the others are not.

**NOTE:** WITH XML POLICIES, REQUEST FILTERS ARE ASSOCIATED WITH THE VIRTUAL DIRECTORIES TAB. WHEN ALL REQUEST FILTERS ON AN XML POLICY ARE DISABLED, THE STATUS OF THE XML POLICY WILL ALSO BE DISABLED (YELLOW STATUS LIGHT).

Requests not matching a defined Request Filter policy will not be processed.

## Request Filter Properties

The following table displays the terms and description of the elements of the Request Filter Properties screen:

TERM	DEFINITION
Name	The name given to the Request Filter.
Format	The following formats are available for Request Filters: <ul style="list-style-type: none"><li>• Simple</li><li>• Web Form</li><li>• Multipart</li><li>• DIME (Direct Internet Message Encapsulation)</li><li>• Web Form Data</li><li>• Streaming</li></ul>

- REST
- MTOM

Note: for JSON use Simple or use the default JSON Request Filter.

Description	A description for the Request Filter.
Identification Expression	An expression using “request filter” syntax, used to match HTTP request to process with this filter.
Parameter	For “Web Form” and “Web Form Data” request filters, the name of the HTML form parameter which contains the data to process.
Convert Content-encoding	<ul style="list-style-type: none"> <li>• The No conversion option means that whatever compression (i.e. HTTP Transfer-encoding) was received from the client (compress, gzip, deflate, or none) will be retained and used for forwarding the XML message to the back end server.</li> <li>• The identity (uncompressed) option means that any compression used by the originating client will be removed before forwarding the uncompressed XML message to the back end server.</li> <li>• The gzip option means that whatever compression was received from the client (compress, gzip, deflate, or none) will be replaced with gzip compression before forwarding the XML message to the back end server.</li> <li>• The deflate option means that whatever compression was received from the client (compress, gzip, deflate, or none) will be replaced with deflate compression before forwarding the XML message to the back end server.</li> </ul>

## Request Filters Available to All XML Policies

The collection of common default request filters on the system is accessed from the **XML Policies** screen, under **Virtual URI Settings**. These request filters affect and apply only to newly created XML policies and represent the collection of all Request Filters available to any newly created Virtual Directory. To view the Request Filters, click **Edit** where you see the Request Filter Policy. The Request Filter Policy screen displays the three enabled request filters.

### REQUEST FILTER POLICIES > REQUEST FILTER POLICY

REQUEST FILTER POLICY					
Policy Name*:		Default			
<input type="checkbox"/>	#	MESSAGE TYPE FILTER	FORMAT	DESCRIPTION	STATUS
<input type="checkbox"/>	1	<a href="#">XML Default</a>	Simple	Plain XML	<span style="color: green;">●</span>
<input type="checkbox"/>	2	<a href="#">HTTP GET</a>	Simple	HTTP GET	<span style="color: green;">●</span>
<input type="checkbox"/>	3	<a href="#">Multipart</a>	Multipart	WSDL 1.1 MIME Filter	<span style="color: red;">●</span>
<input type="checkbox"/>	4	<a href="#">DIME</a>	DIME	WS-Attachments	<span style="color: red;">●</span>
<input checked="" type="checkbox"/>	5	<a href="#">Streaming</a>	Streaming	Generic	<span style="color: green;">●</span>
<input type="checkbox"/>	6	<a href="#">MTOM</a>	MTOM	SOAP Message Transmission Optimization Mechanism	<span style="color: red;">●</span>
<input type="checkbox"/>	7	<a href="#">SMTP Text</a>	Simple	HTML or Plain Text Email	<span style="color: red;">●</span>
<input type="checkbox"/>	8	<a href="#">SMTP MIME</a>	Multipart	Email with Attachments	<span style="color: red;">●</span>

[Enable](#) [Disable](#) [New](#) [Delete](#) [Update](#) [Save](#) [Restore Defaults](#)



## Request Filters Available to Each Virtual Directory

Local default request filters on the system are accessed from the **XML Policies** screen, after selecting an **individual XML Policy name link**. On the Virtual Directory tab, select a **Virtual Directory link**. On the Virtual Directory Details screen, scroll to the middle of the screen to find the Request Filter Policy. Click **Edit** to go the Request Filter Policy screen. These request filters apply only to an individual Virtual Directory on an XML policy and represent the collection of all local Request Filters available to this specific Virtual Directory. The Request Filters area of the screen displays the enabled request filters.

### XML POLICIES

Search Usage: type any text

Filter U

☐ **Virtual Directories**

☐ **Task Lists**

**No Labels**

<input type="checkbox"/>	NAME	VIRTU
<input type="checkbox"/>	<a href="#">New XML Policy</a>	New V
<input type="checkbox"/>	<a href="#">NewXML Policy</a>	New V

<input type="checkbox"/>	<b>VIRTUAL DIRECTORY</b>
<input type="checkbox"/>	<a href="#">New Virtual Directory</a>

### REQUEST FILTER POLICY

Policy Name\*: Default

<input type="checkbox"/>	#	MESSAGE TYPE FILTER	FORMAT	DESCRIPTION	STATUS
<input type="checkbox"/>	1	<a href="#">XML Default</a>	Simple	Plain XML	<span style="color: green;">●</span>
<input type="checkbox"/>	2	<a href="#">HTTP GET</a>	Simple	HTTP GET	<span style="color: green;">●</span>
<input type="checkbox"/>	3	<a href="#">Multipart</a>	Multipart	WSDL 1.1 MIME Filter	<span style="color: red;">●</span>
<input type="checkbox"/>	4	<a href="#">DIME</a>	DIME	WS-Attachments	<span style="color: red;">●</span>
<input type="checkbox"/>	5	<a href="#">Streaming</a>	Streaming	Generic	<span style="color: green;">●</span>
<input type="checkbox"/>	6	<a href="#">MTOM</a>	MTOM	SOAP Message Transmission Optimization Mechanism	<span style="color: red;">●</span>
<input type="checkbox"/>	7	<a href="#">SMTP Text</a>	Simple	HTML or Plain Text Email	<span style="color: red;">●</span>
<input type="checkbox"/>	8	<a href="#">SMTP MIME</a>	Multipart	Email with Attachments	<span style="color: red;">●</span>

[Enable](#) [Disable](#) [New](#) [Delete](#) [Update](#) [Save](#) [Restore Defaults](#)

## Differences between Common and Local Default Request Filters

The following scenario is presented to distinguish between common and local Request Filters. Which event is shown is displayed as either COMMON or LOCAL:

### COMMON

When adding a new Request Filter (**Foo**) to the common collection makes Foo available to any subsequently created Virtual Directories.

**XML POLICIES**

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

No Labels

<input type="checkbox"/>	NAME	VIRTUAL DIRECTORY	STATUS	VIRTUAL URI	R
<input type="checkbox"/>	<a href="#">New XML Policy</a>	<a href="#">New Virtual Directory</a>	<span style="color: green;">●</span>	<a href="http://127.0.0.1:80/n">http://127.0.0.1:80/n</a>	<a href="#">h</a>
<input type="checkbox"/>	<a href="#">NewXMLPolicy</a>	<a href="#">New Virtual Directory</a>	<span style="color: green;">●</span>	<a href="http://127.0.0.1:80">http://127.0.0.1:80</a>	<a href="#">h</a>

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

## REQUEST FILTER POLICIES > REQUEST FILTER POLICY

**REQUEST FILTER POLICY**

Policy Name\*: Default

<input type="checkbox"/>	#	MESSAGE TYPE FILTER	FORMAT	DESCRIPTION	STATUS
<input type="checkbox"/>	1	<a href="#">XML Default</a>	Simple	Plain XML	<span style="color: green;">●</span>
<input type="checkbox"/>	2	<a href="#">HTTP GET</a>	Simple	HTTP GET	<span style="color: green;">●</span>
<input type="checkbox"/>	3	<a href="#">Multipart</a>	Multipart	WSDL 1.1 MIME Filter	<span style="color: red;">●</span>
<input type="checkbox"/>	4	<a href="#">DIME</a>	DIME	WS-Attachments	<span style="color: red;">●</span>
<input type="checkbox"/>	5	<a href="#">Streaming</a>	Streaming	Generic	<span style="color: green;">●</span>
<input type="checkbox"/>	6	<a href="#">MTOM</a>	MTOM	SOAP Message Transmission Optimization Mechanism	<span style="color: red;">●</span>
<input type="checkbox"/>	7	<a href="#">SMTP Text</a>	Simple	HTML or Plain Text Email	<span style="color: red;">●</span>
<input type="checkbox"/>	8	<a href="#">SMTP MIME</a>	Multipart	Email with Attachments	<span style="color: red;">●</span>

[Enable](#) [Disable](#) [New](#) [Delete](#) [Update](#) [Save](#) [Restore Defaults](#)

### LOCAL

You may create other request filter policies by clicking **Add** when in the Request Filter Policies screen. When adding a new Virtual Directory, you can specify the appropriate Request Filter Policy at the Virtual Directory details screen.

Virtual Directories
Task Lists
Settings
IDP Rules
Logging

Virtual Directories > Virtual Directory: New Virtual Directory

VIRTUAL DIRECTORY

Name\*: New Virtual Directory ⓘ  
Description:   
Virtual URI: http://127.0.0.1:80(/.\*)?  
Remote URI: http://www.forumsys.com:80\$0

VIRTUAL URI SETTINGS

Listener Policy: HttpListenerPolicy ▼ [Edit](#)  
Virtual Host:   
☐ Use virtual host as a regular expression  
Virtual Path:   
☐ Enable Virtual Path Case Insensitivity  
Filter Expression: (/.\*)?  
Replace Expression: \$0  
Request Filter Policy: Default ▼ [Edit](#)  
Error Template: [From Listener Policy] ▼

## COMMON

The Request Filter Policy screen shows the built in Default policies as seen below:

REQUEST FILTER POLICIES

14 items found. Search , max results  [Show](#)

☐ REQUEST FILTER POLICY

☐ [Default](#)  
☐ [Default\\_HTML](#)  
☐ [Default\\_JSON](#)  
☐ [Default\\_OAuth](#)  
☐ [Default\\_REST](#)  
☐ [Default\\_STS](#)  
☐ [Request Filter Policy](#)  
☐ [Request Filter Policy-2](#)  
☐ [Request Filter Policy-3](#)  
☐ [Request Filter Policy-4](#)  
☐ [Request Filter Policy-5](#)  
☐ [Request Filter Policy-6](#)  
☐ [Request Filter Policy-7](#)  
☐ [Request Filter Policy-8](#)

[New](#)
[Delete](#)
[Copy](#)
[Restore Defaults](#)

Selecting **Restore Defaults** at the Request Filter Policy screen reverts all the Default Request Filters to their original state. Within each Request Filter there is also a **Restore Defaults** that reverts the particular Request Filter to its original state.

## REQUEST FILTER POLICIES > REQUEST FILTER POLICY

REQUEST FILTER POLICY					
Policy Name*:		Default			
<input type="checkbox"/>	#	MESSAGE TYPE FILTER	FORMAT	DESCRIPTION	STATUS
<input type="checkbox"/>	1	<a href="#">XML Default</a>	Simple	Plain XML	
<input type="checkbox"/>	2	<a href="#">HTTP GET</a>	Simple	HTTP GET	
<input type="checkbox"/>	3	<a href="#">Multipart</a>	Multipart	WSDL 1.1 MIME Filter	
<input type="checkbox"/>	4	<a href="#">DIME</a>	DIME	WS-Attachments	
<input type="checkbox"/>	5	<a href="#">Streaming</a>	Streaming	Generic	
<input type="checkbox"/>	6	<a href="#">MTOM</a>	MTOM	SOAP Message Transmission Optimization Mechanism	
<input type="checkbox"/>	7	<a href="#">SMTP Text</a>	Simple	HTML or Plain Text Email	
<input type="checkbox"/>	8	<a href="#">SMTP MIME</a>	Multipart	Email with Attachments	
<a href="#">Enable</a> <a href="#">Disable</a> <a href="#">New</a> <a href="#">Delete</a> <a href="#">Update</a> <a href="#">Save</a> <a href="#">Restore Defaults</a>					

### Content Types for Request Filters

HTTP requests contain a content-type header field that describes the data contained in the body of the message by means of an Internet media type (content type/subtype). Internet content types are also referred to simply as content types or as MIME types when used as part of a Multimedia Internet Message Extensions (MIME) email message. The content types supported in the system and pre-configured in the product include:

- text/xml
- application/xml
- application/dime
- application/x-www-form-urlencoded
- multipart/form-data
- multipart/related
- application/json

### Type Definitions

The following section describes XML content types supported by the system:

#### The text/xml Media Type

An XML document labeled as text/xml might contain declarations, style sheet-linking processing instructions (PIs), schema information or other declarations that are used to process the document. The default request filters associate text/xml documents with the Simple format, meaning no special conversion will be applied.

#### The application/xml Media Type

An XML document labeled as application/xml might also contain declarations, style sheet-linking processing instructions (PIs), schema information or other declarations that are used to process the document. In the case of application/xml documents, the same default request filter processes these as text/xml documents.

#### The application/dime Media Type

Direct Internet Message Encapsulation (DIME) is a lightweight, binary encapsulation format that can be used to encapsulate multiple application defined entities or payloads of arbitrary type and size into a single message construct. The only parameters described by DIME are the payload type, the length, and an optional payload identifier. Either a URI or a registered media type and the length by an integer indicating the number of octets of the payload identify the type. The optional payload identifier is in the

form of a URI enabling cross-referencing between payloads. The format is strictly an encapsulation format and provides no concepts of a connection or logical circuit and does not address head-of-line problems. It is designed to make as few assumptions about the underlying or encapsulating protocol as possible.

### **The application/x-www-form-URLencoded Media Type**

The application/x-www-form-URLencoded media type means that the variable name-value pairs will be encoded the same way a URL is encoded.

Any special characters, including punctuation, will be encoded as %*nn* where *nn* is the ASCII value for the character in hex. The Web Form default request filter associates application/x-www-form-URLencoded documents with the Web Form format, meaning the URL encoding will be removed prior to processing and re-applied prior to sending the document on to the back end servers.

### **The multipart/form-data Media Type**

The content type multipart/form-data is used by browser submitting forms containing files, non-ASCII data and binary data. The content "multipart/form-data" follows the rules of all multipart MIME data streams.

A "multipart/form-data" message contains a series of parts, each representing a successful control. The parts are sent to the processing agent in the same order the corresponding controls appear in the document stream. Part boundaries should not occur in any of the data; how this is done lies outside the scope of this specification.

As with all multipart MIME types, each part has an optional "Content-Type" header that defaults to "text/plain". User agents should supply the "Content-Type" header, accompanied by a "charset" parameter.

Each part is expected to contain:

- A "Content-Disposition" header whose value is "form-data".
- A name attribute specifying the control name of the corresponding control.

### **The multipart/related Media Type**

The multipart/related media type is intended for compound objects consisting of several inter-related body parts that exist within a component's encapsulated structure. This document defines Multipart/Related content-type and provides examples of its use.

The relationships among the body parts of a compound object distinguish it from other object types. Within a single operating environment the links are often file names. Such links may be represented within a MIME message using content-IDs or the value of some other "Content-" headers.

### **The application/json Media Type**

The application/json media type is intended for JSON messages to be processed by the system. The default request filters associate application/json documents with the Simple format, meaning no special conversion will be applied.

**NOTE: FOR MORE INFORMATION ON REQUEST FILTERS, REFER TO APPENDIX A - REQUEST FILTER BACKGROUND.**

## Common Default Request Filters with XML Policies

A summary of the common default Request Filters that come pre-configured with XML policies are:

REQUEST FILTER NAME	FORMAT	CONTENT TYPES
XML Default	Simple	<ul style="list-style-type: none"><li>• text/xml</li><li>• application/xml</li></ul>
Web Form	Web Form	<ul style="list-style-type: none"><li>• application/x-www-form-urlencoded</li></ul>
Web Form Data	Web Form Data	<ul style="list-style-type: none"><li>• multipart/form-data</li></ul>
HTTP GET	Simple	<ul style="list-style-type: none"><li>• text/xml</li><li>• application/xml</li></ul>
Multipart	Multipart	<ul style="list-style-type: none"><li>• multipart/related</li></ul>
DIME	DIME	<ul style="list-style-type: none"><li>• application/dime</li></ul>
Streaming	Streaming	<ul style="list-style-type: none"><li>• (agnostic)</li></ul>
REST	REST	<ul style="list-style-type: none"><li>• (agnostic)</li></ul>
MTOM	MTOM	<ul style="list-style-type: none"><li>• multipart/related.type=application/xop+xml</li></ul>
JSON	Simple	<ul style="list-style-type: none"><li>• application/json</li></ul>

**Note:** Add a new Request Filter by navigating to the **Virtual Directories** tab, and then click **New** from the HTTP REQUEST FILTER section of the screen. Enter **values**, and then click **Save**.

## Request Filter Syntax

The following table displays literal Request Filter syntax conventions used when creating an identifying expression for a Request Filter:

LITERAL CONVENTION	DEFINITION
	Or
&&	And
( )	Grouping
==	Exact match
==i	Case insensitive (Header field will be matched without regard to case.)
==~	Regular expression match (Header field will be matched to a regular expression or a wild card.)
“ ”	Quotes must surround the value to match.

**Note:** If your business processes use only the default Request Filters, then there is no need to create new Request Filters. Adding a new Request Filter is a global operation, and doing so makes all content types listed in the Request Filter screen available to all documents that are processed on the system.

For information on enabling / disabling or editing a Request Filters, refer to the Common Operations of the *Forum Systems Sentry™ Version 9 Web-based Administration Guide*.

## View or Restore Common Default Request Filters for XML Documents

### Viewing Common Default Request Filters for XML Documents

The common default Request Filters for XML policies can be viewed by navigating to the **XML Policies** screen, and selecting **Settings**. The REQUEST FILTERS screen appears. If any of the common default request filters have been edited or removed, you may restore them back to their factory state by following these steps:

These common default Request Filters are available to all Virtual Directories of all XML policies on the system.

**Note:** When restoring default Request Filters, all previously created Request Filters will be deleted.

### Add a Web Form Request Filter

This instruction displays adding a Web Form request filter:

#### REQUEST FILTER POLICIES > REQUEST FILTER POLICY > MESSAGE TYPE FILTER

**HTTP REQUEST FILTER**

Name\*:

InvoiceWebForm

Format:

Web Form ▼

Description:

Web Form for Invoices

Identification Expression\*:

Content-Type == "application/x-www-form-urlencoded" && method == "POST"

☐ Generate Expression

Methods:

☐ GET

☐ POST

☐ HEAD

☐ PUT

☐ DELETE

☐ OPTIONS

☐ TRACE

☐ CONNECT

Content Types:

☐ ANY

☐ XML

☐ SOAP 1.1

☐ SOAP 1.2

☐ SwA

☐ MIME

☐ MTOM

☐ DIME

☐ JSON

☐ URL Encoded

☐ Web Form

Parameter:

Remote Convert Content-Encoding:

[No conversion] ▼

Client Convert Content-Encoding:

[No conversion] ▼

Create

- From the Navigator, select **Request Filters**. The policy opens with the list of existing request filters displayed.
- Click on the request filters you want to add to, and then select **New**.
- On the REQUEST FILTER details screen, enter a **Request Filter** name in the Name field.
- From the Format drop down list, click **Web Form**.
- Enter a **Description** in the Description field (optional).

**Note:** Review the previous section entitled Request Filter Syntax or enter an identifying expression that parallels the examples below:

Example #1     `Content-Type == "application/x-www-form-urlencoded" && method == "POST"`

Example #2     `( Host == "acme3.com" || Content Type ==~ "acme3/.*" ) && method == "POST"`



You may type either expression into the Identification Expression field, or paste an expression into it.

- Enter an expression that tests HTTP header values in the Identification Expression field. Enter:  
`Content-Type == "application/x-www-form-urlencoded" && method == "POST"`
- Enter **DOCUMENT** (the name of the text field from the posted form) in the Parameter field.
- Skip the Convert Content-Encoding drop down list.
- Click **Create**.

## Promote or Demote a Request Filter Priority

Follow these steps to promote a Request Filter priority. This instruction promotes the Web Form Request Filter:

<input type="checkbox"/>	#	MESSAGE TYPE FILTER	FORMAT	DESCRIPTION	STATUS
<input type="checkbox"/>	1	<a href="#">XML Default</a>	Simple	Plain XML	
<input type="checkbox"/>	2	<a href="#">HTTP GET</a>	Simple	HTTP GET	
<input type="checkbox"/>	3	<a href="#">Multipart</a>	Multipart	WSDL 1.1 MIME Filter	
<input type="checkbox"/>	4	<a href="#">DIME</a>	DIME	WS-Attachments	
<input type="checkbox"/>	5	<a href="#">Streaming</a>	Streaming	Generic	
<input type="checkbox"/>	6	<a href="#">MTOM</a>	MTOM	SOAP Message Transmission Optimization Mechanism	
<input type="checkbox"/>	7	<a href="#">SMTP Text</a>	Simple	HTML or Plain Text Email	
<input type="checkbox"/>	8	<a href="#">SMTP MIME</a>	Multipart	Email with Attachments	
<input type="checkbox"/>	9	<a href="#">InvoiceWebForm</a>	Web Form	Web Form for Invoices	

**Enable   Disable   New   Delete   Update   Save   Restore Defaults**

- From the Navigator, select the **Request Filters** screen then select the specific Request Filter.
- With your mouse, select the **UP arrow** aligned with the Web Form Request Filter.
- The REQUEST FILTERS screen refreshes and the Web Form Request Filter has been promoted.

## Delete a Request Filter

Follow these steps to delete a Request Filter:

- From the Navigator, select the **Request Filters** screen then select the specific Request Filter.
- Checks the checkbox aligned with a Request Filter, and then select **Delete**.

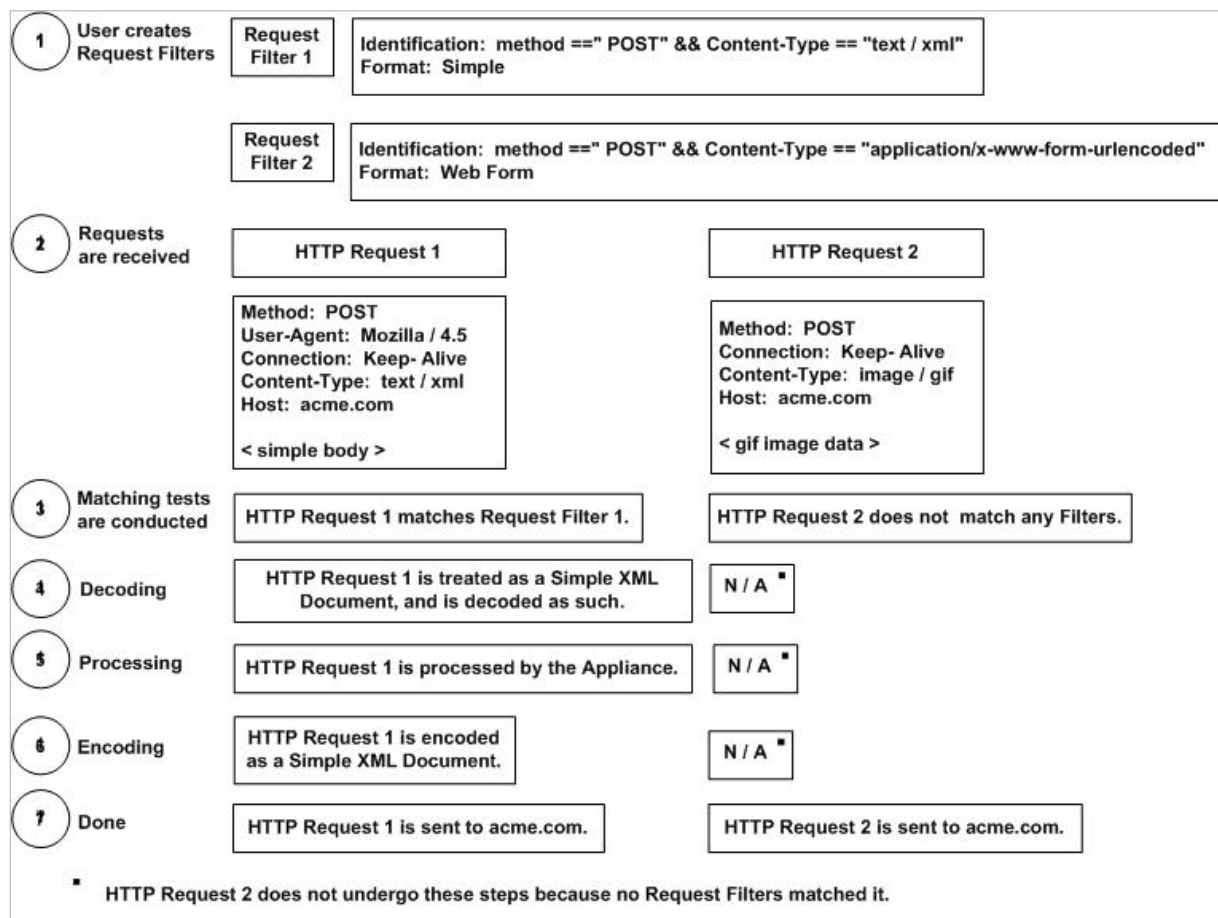


- The “Are you sure that you want to permanently delete all existing filters?” message appears. Click **OK**.

## APPENDIX

### Appendix A - How Request Filters Work

Request Filters identify and decode XML documents of different types as they are prepared for processing in the system, before actual document manipulation. The graphic below displays the actions that occur as Request Filters are applied to a document:



**NOTE:** THIS GRAPHIC ASSUMES THAT THE NO MATCHING XML IDP RULE IS OFF.

Figure 3: Request Filters Identify and Convert XML Documents

## Appendix B - How to Invoke a Request

The product is used for securing XML Web Service requests that are destined for a remote server. The system can act as a security intermediary to HTTP or FTP requests. The payload can be any XML message, including SOAP envelopes that are part of a Post request. HTTP requests can originate from a web browser or HTTP applications. Before an HTTP request can be intercepted by the system, Network policies would have been created in the product with both the listener IP address and port number and remote server IP address and port number defined.

A Network policy is a set of business rules that identifies a listener. Listeners are part of an extensible framework that handles requests and messages for XML security. Network policy names reflect the traffic of systems in your organization that are affected by these business rules.

When your customers create an HTTP request, they are sending an XML Document to the system. From the method (POST) and request filter type (For example, URLEncoded or Simple (also referred to as Single Part), the system intercepts and identifies the request. When there is a match, the request is processed by a specific Network policy.

### Request Examples

The following are three requests, each sent differently, but containing the same data.

#### Request 1 - URL-Encoded Request

Request #1 is from an HTTP client using the POST method and the *application/x-www-form-urlencoded* Media / MIME type:

```
POST / HTTP/1.0
Content-Length: 946
Content-Type: application/x-www-form-urlencoded
```

```
DOCUMENT=%3CInvoice%3E%20%20%20%20%0A%09%3CInvoiceNo%3E12%3C%2FInvoiceNo%3E%20%20%20%20%0A%09%3COrderDate%3E1996%2D07%2D04T00%3A00%3A00%3C%2FOrderDate%3E%20%20%20%20%20%20%20%0A%09%09%3CQuantity%3E5%3C%2FQuantity%3E%20%20%20%20%0A%09%3C%2FItem%3E%0A%3C%2FInvoice%3E%0A%0A
```

#### Request 2 - APPLICATION/XML Request

Request #2 is from an HTTP client using the POST method and the *application/xml* Media / MIME type:

```
POST / HTTP/1.0
Content-Length: 473
Content-Type: application/xml

<Invoice>
  <InvoiceNo>12</InvoiceNo>
  <OrderDate>1996-07-04T00:00:00</OrderDate>
  <Item>
    <SystemID>11</SystemID>
    <Price>14</Price>
    <Quantity>12</Quantity>
  </Item>
  <Item>
    <SystemID>42</SystemID>
    <Price>9</Price>
    <Quantity>10</Quantity>
  </Item>
  <Item>
    <SystemID>72</SystemID>
```

```

        <Price>34.8</Price>
        <Quantity>5</Quantity>
    </Item>
</Invoice>

```

### Request 3 - Browser to System

The following is an HTML web page that a client could use to submit a request to a remote server at IP address 10.5.3.90, Port 7100 where the end application is a Java servlet called XMLSimpleServlet. To the system, this servlet is actually the remote server. Request #3 is from an HTTP browser using the POST method and the *application/x-www-form-urlencoded* Media / MIME type:

```

<html>
<body bgcolor="#505050">
<center>
<hr>
<form method="POST"
action="http://10.5.3.90:7100/servlet/XMLSimpleServlet/document_posted"
target="frame2">
<input type="image" src="submit.gif" alt=="Submit XML Data">
<br><hr><br>
<textarea name="DOCUMENT" rows=20 cols=60>
<Invoice>
    <InvoiceNo>12</InvoiceNo>
    <OrderDate>1996-07-04T00:00:00</OrderDate>
    <Item>
        <SystemID>11</SystemID>
        <Price>14</Price>
        <Quantity>12</Quantity>
    </Item>
    <Item>
        <SystemID>42</SystemID>
        <Price>9</Price>
        <Quantity>10</Quantity>
    </Item>
    <Item>
        <SystemID>72</SystemID>
        <Price>34.8</Price>
        <Quantity>5</Quantity>
    </Item>
</Invoice>
</textarea>
<br><br><hr>
</form>
</center>
</html>

```

## Appendix C - Sample Request Filters

### Sample Simple Request Filter Expression

The following expression is a Simple Request Filter that identifies a *content type* that matches *text/xml* or *application/xml* and uses *POST* as its method.

```
( Content-Type == "text/xml" || Content-Type == "application/xml" )  
&& method == "POST"
```

### Sample Web Form Request Filter Expression

The following expression is a Web Form Request Filter that identifies a *content type* that matches *application/x-www-form-urlencoded* and uses *POST* as its method.

```
Content-Type == "application/x-www-form-urlencoded" && method ==  
"POST"
```

### Sample HTTP GET Request Filter Expression

The following expression is an HTTP GET Request Filter that identifies a *content type* that matches *text/xml* or *application/xml* and uses *HTTP GET* as its method.

```
( Content-Type == "GET" || Content-Type == "application/xml" ) &&  
method == "GET"
```

### Sample Multipart Request Filter Expression

The following expression is a Multipart Request Filter that identifies the properties of a Request Filter that includes an outer *content type* that matches *multipart/related* and that includes a primary inner content type of *text/xml* and uses *POST* as its method.

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

### Sample DIME Request Filter Expression

The following expression is a DIME Request Filter that identifies a *content type* that matches *application/dime* and uses *POST* as its method.

```
(Content-Type ==i "application/dime") && (method == "POST")
```

### Sample Web Form Data Request Filter Expression

The following expression is a Web Form Data Request Filter that identifies a *content type* that matches *multipart/form-data* and uses *POST* as its method.

```
(Content-Type ==i "multipart/form-data") && (method == "POST")
```

### Sample Streaming Request Filter Expression

The following expression is a Streaming Request Filter that identifies a *content type* that matches any content type that uses *POST* as its method.

```
(Content-Type ==~ ".*") && (method == "POST")
```

## Appendix D - Constraints in XML Policies Guide

ELEMENT	CONSTRAINTS	CHARACTER COUNT
XML policy Names	Unique and case sensitive. Must start with an alpha character. Accepts underscores and dashes.	1-32
Virtual Directory name	Unique and case sensitive	1-256
Request Filter name	Unique and case sensitive	1-256

## Appendix E - Specifications in XML Policies Guide

ELEMENT SUPPORTED	SPECIFICATIONS
XML policies	Unlimited *
Virtual Directories	With XML policies, you may have an unlimited number of Virtual Directories per XML policy.
Request Filters	100
Task Lists allowed per XML policy	Unlimited * Task Lists are associated to Task List Groups, not directly to XML Policies. Task List Groups can contain multiple Task List.
Task List Groups allowed per XML policy	1 Task List Group can be set at the following levels: <ul style="list-style-type: none"><li>• Virtual Directory for Requests</li><li>• Virtual Directory for Responses</li><li>• XML Policy for Requests</li><li>• XML Policy for Responses</li></ul>

\* Limited only by disk space.

## Appendix F - Virtual Directory Reference Chart in XML Policies Guide

Click on the Virtual Directory name link to view available options in a Virtual Directory.

The screenshot shows the 'Virtual Directories' configuration interface. At the top are tabs for 'Virtual Directories', 'Task Lists', 'Settings', and 'IDP Rules'. The main heading is 'Virtual Directories > Virtual Directory: New Virtual Directory'. Below this is a form with various fields and checkboxes. Red arrows point from explanatory text on the right to specific fields in the form. Blue arrows point from the same text to other fields. A table at the bottom lists HTTP request filters.

**VIRTUAL DIRECTORY**

Name\*: New Virtual Directory

Description:

Listener Policy: Bayside\_Listener

Virtual Path: /virtual/service

Virtual URI: https://10.5.6.92:8034/virtual/service/?

Filter Expression: /?

Replace Expression: \$0

☒ Send to remote server

☐ Discard response from server

Remote Policy: Bayside\_Remote

Remote Path: /remote

Remote URI: http://www.server.com:8080/remote/\$0

Process Response: On

ACL: EastCoast\_ACL

Error Template: [From Listener Policy]

From the Listener Policy drop down list, select a Listener Policy to associate with this XML Policy.

The Virtual Path field allows users to customize this XML policy's Virtual Path.

With **Send to remote server** checked, the Remote Policies drop down list becomes enabled.

With **Discard response from server** checked, any responses from the back end server are discarded.

From the Remote Policies drop down list, select a **Remote Policy** to associate with this XML Policy.

The Remote Path field allows users to customize this XML policy's Remote Path.

From the Access Control List drop down, select an **ACL Policy** to enforce on this XML policy. The "Allow All" ACL means there is no access control enforced.

From the Error Template drop down list, select the **Error Template Policy** referenced on the Listener policy, or select another one.

#	HTTP REQUEST FILTER	FORMAT	DESCRIPTION	STATUS
1	XML_Default	Simple	Plain XML	●
2	Web_Form	Web Form	Posted form (URL Encoded)	●
3	HTTP_GET	Simple	HTTP GET	●
4	Multipart	Multipart	SOAP with Attachments	●
5	DIME	DIME	WS-Attachments	●

Select a **Request Filter** link to view details, or select **New** to create a new HTTP Request Filter.

Buttons: Restore Defaults, Enable, Disable, Delete, New

Figure 4: The Virtual Directories Screen and Associated Options with XML Policies.

# INDEX

- add a Web Form Request Filter, 31
- add a XML policy while creating a Listener policy, 8
- add a XML policy while creating a Remote policy, 8
- add XML policy and associate existing Listener, 7
- application/dime media type, 28
- application/xml media type, 28
- application/x-www-form-urlencoded media type, 29
- common default Request Filters, 30
  - restoring, 31
  - viewing, 31
- common Request Filter, 24
- communication mode
  - Proxy mode, 14
  - Service mode, 14
- content type header, 28
- content types supported for Request Filters, 28
- content-encoding conversion options with request filters, 24
- conventions used, 4
- convert content-encoding options with request filters, 24
- default Filter Expression, 15
- deflate
  - content-encoding conversion option with request filters, 24
- delete Request Filter, 32
- description of Virtual Directory of an XML policy, 10, 19
- DIME Request Filter expression
  - example of, 37
- Discard response from server of Virtual Directory of an XML policy, 11
- Discard response from server on Virtual Directory, 39
- Error Template of Virtual Directory of an XML policy, 12
- examples for XML policy, 7
- export XML policies, 21
- expression that tests HTTP headers
  - Media Type, 32
- Filter Expression
  - default in WSDL policy, 15
- Filter Expression of Virtual Directory of an XML policy, 11
- fs\_password, 15
- fs\_user, 15
- gzip
  - content-encoding conversion option with request filters, 24
- HTTP GET Request Filter expression
  - example of, 37
- identity (uncompressed)
  - content-encoding conversion option with request filters, 24
- IDP Rules tab terms, 20
- import XML policies, 21
- Listener Policy of Virtual Directory of an XML policy, 10
- local Request Filter, 25
- media type
  - application/dime, 28
  - application/xml, 28
  - application/x-www-form-urlencoded, 29
  - expression that tests HTTP headers, 32
  - multipart/form-data, 29
  - multipart/related, 29
  - text/xml, 28
- media type definitions, 28
- MIME types, 28
- mix protocols on an XML policy, 14
- Multipart Request Filter expression
  - example of, 37
- multipart/form-data media type, 29
- multipart/related media type, 29
- name of Virtual Directory of an XML policy, 10, 19
- no conversion
  - content-encoding conversion option with request filters, 24
- parameter
  - for Web Form Request Filter format, 32
- Process Response of Virtual Directory of an XML policy, 11
- promote / demote Request Filter priority, 32
- Protect virtual resource
  - Settings tab, 19
- protocol mixing on an XML policy, 14
- Proxy mode
  - communication mode, 14
- Remote Path of Virtual Directory of an XML policy, 11
- Remote Policy of Virtual Directory of an XML policy, 11
- Remote URI of Virtual Directory of an XML policy, 11



- Replace Expression of Virtual Directory of an XML policy, 11
- Request Filter
  - adding a Web Form, 31
  - common, 24
  - deleting, 32
  - format for Web Form, 32
  - local, 25
  - promoting/demoting priority, 32
  - syntax, 30
- Request Filters
  - common default, 30
  - content types supported, 28
  - how they work, 34
- Request Filters in XML policies, 28
- restore common Default Request Filters, 31
- Send to remote server of Virtual Directory of an XML policy, 11
- Service mode
  - communication mode, 14
- Settings tab in XML policy, 19
- Simple Request Filter expression
  - example of, 37
- terms
  - in IDP Rules tab, 20
  - on Virtual Directories tab for XML policy, 10
  - on Virtual Directory of an XML policy, 10
- text/xml media type, 28
- transfer XML policies, 21
- use existing Listener policy for XML policy, 9
- User ACL of Virtual Directory of an XML policy, 11
- view common Default Request Filters, 31
- Virtual Directories tab in XML policy, 9
- Virtual Directories tab screen terms, 10
- Virtual Directory
  - description, 10, 19
  - Discard response from server, 11, 39
  - Discard send to remote server, 11
  - Error Template, 12
  - Filter Expression, 11
  - Listener Policy, 10
  - name, 10, 19
  - Process Response, 11
  - Remote Path, 11
  - Remote Policy, 11
  - Remote URI, 11
  - Replace Expression, 11
  - User ACL, 11
  - Virtual Path, 10
  - Virtual URI, 10
- Virtual Directory terms, 10
- Virtual Path of Virtual Directory of an XML policy, 10
- Virtual URI of Virtual Directory of an XML policy, 10
- Web Form
  - Request Filter format, 32
- Web Form Request Filter expression
  - example of, 37
- Web Form Request Filter format
  - parameter for, 32
- WSDL policy
  - default Filter Expression, 15
- XML policy, 6
  - adding and associate existing Listener, 7
  - adding while creating a Remote policy, 8
  - adding while creating Listener policy, 8
  - examples, 7
  - mixing protocols on an XML policy, 14
  - Proxy mode, 14
  - Service mode, 14
  - Settings tab, 19
  - using existing Listener policy, 9
  - Virtual Directories tab, 9
- XML Policy names, 7, 8