



# **FORUM SENTRY™ VERSION 9**

## **JSON 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 JSONSec™ WebAdmin, Forum Systems JSON 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 JSON Policies Guide, published May 2024.

D-ASF-SE-017698

## Table of Contents

INTRODUCTION TO THE JSON POLICIES GUIDE .....	4
Audience for the JSON Policies Guide .....	4
Conventions Used in the JSON Policies Guide .....	4
JSON POLICIES .....	6
JSON Features .....	6
JSON Policy Examples .....	7
VIRTUAL DIRECTORIES .....	9
Virtual Directories Tab Screen Terms for JSON Policies .....	9
Virtual Directory Detail Terms for JSON Policies .....	9
Operations on Virtual Directories for JSON Policies .....	13
Processing in Proxy and Service Modes .....	13
Protocol Mixing with JSON Policies .....	14
Default Filter Expression in a Virtual Directory .....	15
TASK LISTS AND TASK LIST GROUPS FOR JSON POLICIES .....	16
Task Lists Groups at the Virtual Directory Level .....	16
Task Processing at the JSON Policy Level .....	17
SETTINGS FOR JSON POLICIES .....	18
IDP RULES FOR JSON POLICIES .....	19
IDP Rule Tab Screen Terms for JSON Policy .....	19
LOGGING SETTINGS FOR JSON POLICIES .....	19
Logging Tab Screen Terms for JSON Policy .....	19
TRANSFERRING EXPORTING AND IMPORTING JSON POLICIES .....	20
REQUEST FILTERS FOR JSON POLICIES .....	21
Request Filters Available to All JSON Policies .....	22
Request Filters Available to Each Virtual Directory .....	22
Content Types for Request Filters .....	23
Common Default Request Filters with JSON Policies .....	23
Request Filter Syntax .....	24
APPENDIX .....	25
Appendix A - How Request Filters Work .....	25
Appendix B - Constraints in JSON Policies Guide .....	26
Appendix C - Specifications in JSON Policies Guide .....	26
Appendix D - Virtual Directory Reference Chart in JSON Policies Guide .....	27
INDEX .....	28

## List of Figures

Figure 1: Virtual Directories Status .....	5
Figure 2: Enable JSON Filter .....	5
Figure 3: Virtual Directory Details .....	6
Figure 4: Virtual Directory Creation .....	6
Figure 5: New JSON Policy .....	7
Figure 6: Proxy and Service Modes .....	13
Figure 7: Protocol Mixing on JSON Policies .....	14
Figure 8: Task Processing .....	17
Figure 9: Request Filter WorkFlow .....	25
Figure 10: The Virtual Directories Screen .....	27

# INTRODUCTION TO THE JSON POLICIES GUIDE

## Audience for the JSON Policies Guide

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

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

## 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 a JSON policy, refer to the *Forum Systems Sentry™ Version 9 Tasks Management Guide*.

## Conventions Used in the JSON 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 JSON 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 JSON policies, and Virtual Directories.



Virtual Directories		
<div>Task Lists</div> <div>Settings</div>		
<input type="checkbox"/>	<b>VIRTUAL DIRECTORY</b>	<b>STATUS</b>
<input type="checkbox"/>	<a href="#">New Virtual Directory</a>	
<input type="checkbox"/>	<a href="#">New Virtual Directory2</a>	

Figure 1: Virtual Directories Status

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

<input type="checkbox"/>	#	MESSAGE TYPE FILTER	FORMAT	DESCRIPTION	STATUS
<input type="checkbox"/>	1	<a href="#">JSON</a>	Simple	JSON	
<div><div>Restore Defaults</div><div>Enable</div><div>Disable</div><div>Delete</div><div>New</div></div>					

Figure 2: Enable JSON Filter

## JSON POLICIES

A JSON policy is a set of rules that provide a policy for processing of JSON flowing through the system.

JSON policies include the following accessible properties and actions; each of which manage a portion of the JSON policy and are detailed later:



The screenshot shows the 'Virtual Directories' tab selected. The breadcrumb path is 'Virtual Directories > Virtual Directory: New Virtual Directory'. The form is titled 'VIRTUAL DIRECTORY' and contains the following fields:

- Name\*:** A text input field containing 'New Virtual Directory'.
- Description:** An empty text input field.
- Listener Policy:** A dropdown menu showing 'NewXMLPolicy-Listener' with an 'Edit' link to its right.
- ☐ **Use virtual host as a regular expression**
- Virtual Host:** An empty text input field.

Figure 3: Virtual Directory Details

- **Virtual Directories:** Manage the services of the JSON 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 JSON 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.

The screenshot shows the 'Virtual Directories' tab with a table listing the created virtual directory. The table has four columns: 'VIRTUAL DIRECTORY', 'STATUS', 'VIRTUAL URI', and 'REMOTE URI'. Below the table are four action buttons: 'Enable', 'Disable', 'Delete', and 'New'.

 VIRTUAL DIRECTORY	STATUS	VIRTUAL URI	REMOTE URI
 <a href="#">New Virtual Directory</a>		http://10.5.1.14:88/xml	http://10.5.1.17:80/xml

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

Figure 4: Virtual Directory Creation

From an open JSON 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.

## JSON Features

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

- Add a JSON policy.
- Create a new or associate an 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 JSON policy.
- Transfer, import or export JSON policies. (For more information, refer to the *Forum Systems Sentry™ Version 9 System Management Guide*.)

## JSON Policy Examples

Examples for a JSON policy include:

- Add a JSON Policy.
- Create New Network Policies for JSON Policy.
- Use Existing Network Policy for JSON Policy.
- View Virtual Directories of a JSON Policy.

### Add a JSON Policy

When adding a JSON policy, you may associate any existing Listener. Follow these steps to add a JSON policy and associate an existing Listener policy:

### Adding a JSON Policy



JSON POLICIES > NEW JSON POLICY

NEW JSON POLICY

Name\*:

Next

**Figure 5: New JSON Policy**

- Navigate to the **JSON Policies** screen and select **New**. In the Name field, enter the **Name** for this JSON policy.
- In the Description field, enter a **Description** for this JSON 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 JSON Policy

You may create a new Listener Policy when creating a JSON Policy:

## JSON POLICIES > NEW JSON POLICY

### SET LISTENER POLICY

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

☒ Select from existing listener policies

AmqpListenerPolicy

☐ Create a new HTTP listener policy

Listener Policy Name\*: NewJSONPolicy2-Listener

Use Device IP: ☐

Listener IP\*: 192.168.58.1

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

GroupRemotePolicy

☐ Create a new HTTP remote policy for this remote server

Remote Policy Name\*: NewJSONPolicy2-Remote

Remote Policy Host\*:

Remote Policy Port\*: 80

- 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 JSON 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 JSON policy, as well as the Virtual URI and the Remote URI. JSON 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 JSON policy. Each virtual directory is used to map a virtual URI (local) to the physical path and URI (remote, as defined in the JSON document).

**NOTE: WHERE HTTP POLICIES ARE DISCUSSED, NOT ALL OTHER NETWORK POLICIES ARE VALID.**

A Virtual Directory is a pattern which matches an incoming HTTP request URI. A Virtual Directory is defined on the port node in a JSON policy. Because the physical endpoint defined in the JSON 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 JSON Policies

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

TERM	DEFINITION
Virtual Directory	Local URIs used to access the JSON 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 JSON Policies

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

TERM	DEFINITION
Name	The identifier of this Virtual Directory.

TERM	DEFINITION
Description	An optional description of this Virtual Directory.
Virtual URI	The Unique Resource Indicator (URI) path used by clients to access this policy. This is where the system receives a request.
Remote URI	Actual URI back-end server.
Listener Policy	The Listener Policy on the system to associate with this Virtual Directory.
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>
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 Path	The Virtual Path field allows users to customize this JSON's virtual path.
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.
Request Filter Policy	<p>The Request Filter Policy associated with this Virtual Directory. The Request Filter Policy include a list of Request Filter that matches requests based on HTTP headers such as Content-Type or the HTTP method.</p> <p>For information on HTTP Request Filters, refer to the Request Filters for JSON Policies section of this document.</p>
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.
Google Analytics	The Google Analytics policy associated with this Virtual Directory. The Google Analytics policy enables sending data to a Google Analytics account

TERM	DEFINITION
IP ACL Policy	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 Policy	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 ACL Policy grants access of this JSON policy to any member of the User ACL.
XACML Policy	The XACML Policy associated with this Virtual Directory.
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 Kerberos 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> <li>• Password Authentication Realm use in combination with basic authentication</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.
Request Processing	The Task List Group or Task List selected to process the request messages for this Virtual Directory.
Response Processing	The Task List Group or Task List selected to process the response message for this Virtual Directory.
Send to remote server	<ul style="list-style-type: none"> <li>• When checked, the Remote Policies drop down list is enabled. 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>

TERM	DEFINITION
Remote Policy	The Remote Policy associated with this Virtual Directory.
Remote Path	The back-end server path .
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.
Discard response from server	When checked, responses from the back-end server are discarded.

## Operations on Virtual Directories for JSON Policies

JSON 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 a JSON 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.
- Select a Redirect Policy for the Virtual Directory.

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

## Processing in Proxy and Service Modes

Figure 6 displays processing in Proxy or Service modes:



Figure 6: 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 a JSON 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 JSON Policies

Protocol mixing with JSON policies provides a method of mixing protocols between incoming request and outgoing responses on the system.

### How the System Manages Protocol Mixing on JSON Policies

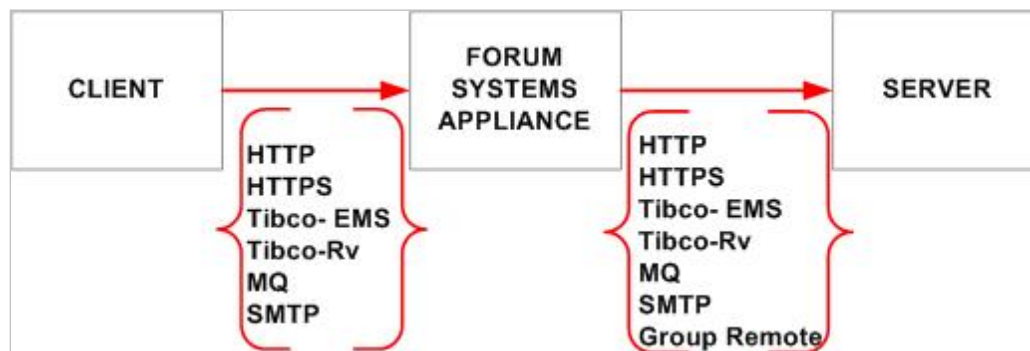


Figure 7: Protocol Mixing on JSON Policies.

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

## Asynchronous Protocols Supported with JSON 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

<https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html>.

**NOTE:** THE DEFAULT FILTER EXPRESSION `"/?"` IS MORE JSONRICTIVE 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 (JSON 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. Others, 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 JSON POLICIES

The Task List tab allows users to view all Tasks and Task Lists associated with a JSON 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 JSON Policy. In previous releases, a single Task List Group was set for all messages (request and response) for the Virtual Directory.

### 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 or Task Lists can be selected for the request or response messages.



## Task Processing at the JSON Policy Level

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

The screenshot displays the 'JSON POLICIES > JSON POLICY' configuration page. Under the 'JSON POLICY' header, the 'Policy Name' is 'New JSON Policy'. A navigation bar includes tabs for 'Virtual Directories', 'Task Lists' (which is active), 'Settings', 'IDP Rules', and 'Logging'. The 'TASK PROCESSES' section contains two main areas: 'Request Processing' and 'Response Processing'. In 'Request Processing', there is a 'Task List Groups' dropdown menu, a 'Type or select label' dropdown menu, and a text field containing 'System Request Group' with an 'Edit' link. Below this is a table with columns 'TASK LIST' and 'STATUS', showing 'No items to display'. The 'Response Processing' section has a 'Task List Groups' dropdown menu, a 'Type or select label' dropdown menu, and a dropdown menu currently set to '[None]'. At the bottom right of the form are 'Create' and 'Save' buttons.

Figure 8: Task Processing.

**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 JSON POLICIES

The Settings tab includes name and description for this JSON 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 JSON Policy.
Policy Description	An optional description of this JSON Policy.
Labels	A label that can be used to group policies that are related.
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>
Authorize based only on the root directory, not the full resource path	When enabled the authentication and authorization is done using the root directory only and not the full resource path.
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 not 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>
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 Resources->WAF Policies
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)

## IDP RULES FOR JSON POLICIES

Intrusion Detection and Prevention (IDP) Rules define a set of criteria, which can be associated with a JSON policy. IDP Groups represent a reusable collection of IDP Rules that may be applied to this JSON 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 JSON 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 JSON POLICIES

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

### Logging Tab Screen Terms for JSON 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 JSON Policy.  When not checked, policy level logging is disabled for the JSON Policy.
Policy Log Level	When policy level logging is enabled, this is the log level set for this policy.
Override log level for the following codes	When enabled, the configured log codes can be excluded or included independently of the original log level for the message code. For example, if message code 00000 is a debug log message, this functionality enables excluding the log message. The message code can be a partial match, 0 will match all codes that start with a 0.
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 JSON POLICIES

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

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

Through the Import / Export screen, users may import JSON 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 a JSON 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 a JSON 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*.
- To Import a JSON 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 6.5 System Management Guide*.

## REQUEST FILTERS FOR JSON 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 JSON 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 JSON policy.

Both sets of Request Filters include:

- JSON

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

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></ul>
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 forms 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 JSON 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 JSON 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 JSON 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 JSON message to the back end server.</li></ul>

## Request Filters Available to All JSON Policies

The collection of common default request filters on the system is accessed from the **JSON Policies** screen, under **Settings**. These request filters affect and apply only to newly created JSON policies and represent the collection of all Request Filters available to any newly created Virtual Directory. The Request Filters area of the screen displays the three enabled request filters.

## Request Filters Available to Each Virtual Directory

Local default request filters on the system are accessed from the **JSON Policies** screen, after selecting an **individual JSON Policy name link**. On the Virtual Directory tab, select a **Virtual Directory link**. On the Virtual Directory Details screen, the request filters are associated through a Request Filter Policy.

## 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.

### LOCAL

You may create other request filters by clicking **Add**, and all added request filters created from this screen are local to this given JSON policy.

## Content Types for Request Filters

HTTP requests contain a content-type header field which 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:

- application/JSON

## Type Definitions

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

### 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.

## Common Default Request Filters with JSON Policies

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

REQUEST FILTER NAME	FORMAT	CONTENT TYPES
JSON Default	Simple	<ul style="list-style-type: none"><li>• application/JSON</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/JSON</li><li>• application/JSON</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>

**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*.



## APPENDIX

### Appendix A - How Request Filters Work

Request Filters identify and decode JSON 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:

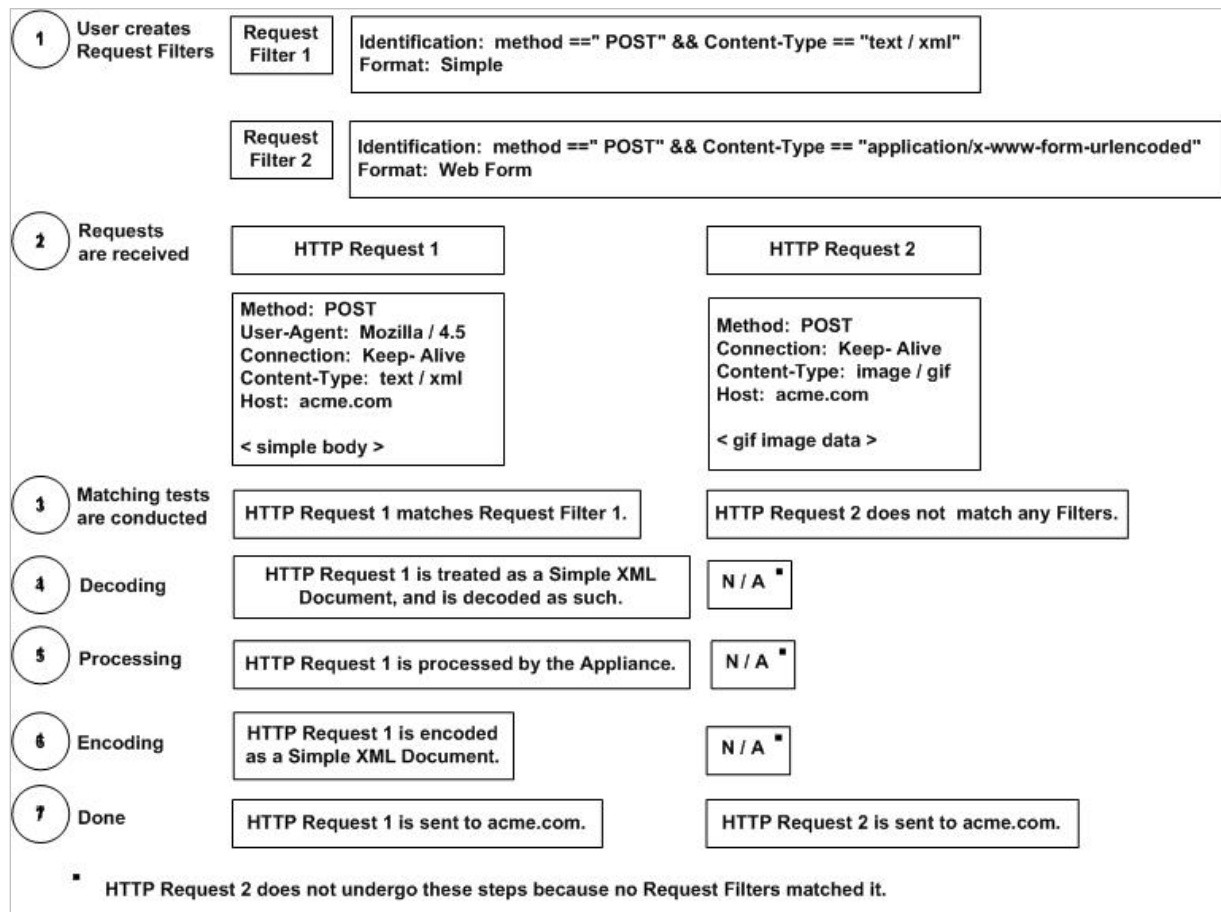


Figure 9: Request Filter WorkFlow.

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

## Appendix B - Constraints in JSON Policies Guide

ELEMENT	CONSTRAINTS	CHARACTER COUNT
JSON 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 C - Specifications in JSON Policies Guide

ELEMENT SUPPORTED	SPECIFICATIONS
JSON policies	Unlimited *
Virtual Directories	With JSON policies, you may have an unlimited number of Virtual Directories per JSON policy.
Request Filters	100
Task Lists allowed per JSON policy	Unlimited * Task Lists are associated to Task List Groups, not directly to JSON Policies. Task List Groups can contain multiple Task Lists.
Task List Groups allowed per JSON 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>• JSON Policy for Requests</li><li>• JSON Policy for Responses</li></ul>

\* Limited only by disk space.

## Appendix D - Virtual Directory Reference Chart in JSON Policies Guide

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

The screenshot shows the 'Virtual Directories' configuration page. 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	<a href="#">XML Default</a>	Simple	Plain XML	●
2	<a href="#">Web Form</a>	Web Form	Posted form (URL Encoded)	●
3	<a href="#">HTTP GET</a>	Simple	HTTP GET	●
4	<a href="#">Multipart</a>	Multipart	SOAP with Attachments	●
5	<a href="#">DIME</a>	DIME	WS-Attachments	●

Restore Defaults Enable Disable Delete New

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

Figure 10: The Virtual Directories Screen.

# INDEX

## A

- add a JSON policy while creating a Listener policy ....7
- add JSON policy and associate existing Listener.....7

## C

- common default Request Filters .....23
- common Request Filter .....22
- content type header .....23
- content types supported for Request Filters.....23
- conventions used.....4

## D

- default Filter Expression .....15
- deflate
  - content-encoding conversion option with request filters.....22

## F

- fs\_user .....15

## G

- gzip
  - content-encoding conversion option with request filters.....22

## I

- identity (uncompressed)
  - content-encoding conversion option with request filters.....22
- IDP Rules tab terms .....19

## J

- JSON policy .....6
  - examples.....7
  - Proxy mode.....14
  - Service mode .....14

## L

- local Request Filter .....22
- Logging
  - Enable Policy Level Loggign Settings.....20
  - Override log level for the following codes.....20
  - Pattern Match Policy .....20
  - Policy Log Level .....20
- Logging Rules tab terms .....19

## M

- media type definitions.....23
- MIME types.....23
- mix protocols on a JSON policy.....14

## N

- no conversion
  - content-encoding conversion option with request filters.....22

## P

- Proxy mode
  - communication mode.....14

## R

- Request Filter
  - syntax.....24
- Request Filters
  - how they work .....25
- Request Filters in JSON policies .....23

## S

- Service mode
  - communication mode.....14
- Settings tab in JSON policy .....18

## T

- Test .....7
- transfer JSON policies .....20

## U

- use existing Listener policy for JSON policy.....9

## V

- Virtual Directories tab in JSON policy .....9
- Virtual Directories tab screen terms.....9
- Virtual Directory
  - ACL Policy.....11
  - Authorized based only on the root directory, not the full resource path .....18
  - description .....10, 18
  - Discard response from server .....12, 27
  - Discard send to remote server .....11
  - Enable Session Cookies.....18
  - Error Template.....10
  - Filter Expression .....10

Google Analytics .....	10
Host Header .....	12
IP ACL Policy.....	11
Labels.....	18
Listener Policy.....	10
name.....	9, 18
Password Authentication.....	11
Process Response .....	12
Protect Virtual Resource.....	18
Redirect Policy .....	11
Remote Path .....	12
Remote Policy .....	12

Remote URI.....	10
Replace Expression .....	10
Request Filter Policy .....	10
Request Processing.....	11
Response Processing .....	11
Use Virtual host as a regular expression.....	10
Virtual Host .....	10
Virtual Path .....	10
Virtual URI.....	10
XACML Policy .....	11
Virtual Directory terms .....	9