



ForeRunner ASN-9000 **Filters Reference Manual**

MANU0280-02 - Rev. A - July 27, 1998

Software Version ASN_FT 5.0.x

FORE Systems, Inc.

1000 FORE Drive
Warrendale, PA 15086-7502
Phone: 724-742-4444
FAX: 724-742-7742

<http://www.fore.com>

Legal Notices

Copyright © 1995-1998 FORE Systems, Inc. All rights reserved. FORE Systems is a registered trademark, and *ForeRunner*, *ForeView*, *ForeThought*, *ForeRunnerLE*, *PowerHub*, and *CellPath* are trademarks of FORE Systems, Inc. All other brands or product names are trademarks of their respective holders.

U.S. Government Restricted Rights. If you are licensing the Software on behalf of the U.S. Government (“Government”), the following provisions apply to you. If the Software is supplied to the Department of Defense (“DoD”), it is classified as “Commercial Computer Software” under paragraph 252.227-7014 of the DoD Supplement to the Federal Acquisition Regulations (“DFARS”) (or any successor regulations) and the Government is acquiring only the license rights granted herein (the license rights customarily provided to non-Government users). If the Software is supplied to any unit or agency of the Government other than DoD, it is classified as “Restricted Computer Software” and the Government’s rights in the Software are defined in paragraph 52.227-19 of the Federal Acquisition Regulations (“FAR”) (or any successor regulations) or, in the cases of NASA, in paragraph 18.52.227-86 of the NASA Supplement to the FAR (or any successor regulations).

Printed in the USA.

No part of this work covered by copyright may be reproduced in any form. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

This publication is provided by FORE Systems, Inc. “as-is” without warranty of any kind, either express or implied, including, but not limited to, the implied warranties or conditions of merchantability or fitness for a particular purpose. FORE Systems, Inc. shall not be liable for any errors or omissions which may occur in this publication, nor for incidental or consequential damages of any kind resulting from the furnishing, performance, or use of this publication.

Information published here is current or planned as of the date of publication of this document. Because we are improving and adding features to our products continuously, the information in this document is subject to change without notice.

RESTRICTED RIGHTS LEGEND. Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 (October 1988) and FAR 52.227-19 (June 1987).

The VxWorks software used in the Mini Loader is licensed from Wind River Systems, Inc., Copyright ©1984-1996.

FCC CLASS A NOTICE

WARNING: Changes or modifications to this unit not expressly approved by the party responsible for compliance could void this user’s authority to operate this equipment.

NOTE: The PowerHub 7000 and PowerHub 8000 have been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15, FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

DOC CLASS A NOTICE

This digital apparatus does not exceed Class A limits for radio noise emission for a digital device as set out in the Radio Interference Regulations of the Canadian Department of Communications.

Le present appareil numerique n’emet pas de bruits radioelectriques depassant les limites applicables aux appareils numeriques de la class A prescrites dans le reglement sur le brouillage radioelectrique edicte par le ministere des Communications du Canada.

VCCI CLASS 1 NOTICE

この装置は、第一種情報処理装置（商工業地域において使用されるべき情報処理装置）で商工業地域での電波障害防止を目的とした情報処理装置等電波障害自主規制協議会(VCCI)基準に適合しております。

従って、住宅地域またはその隣接した地域で使用すると、ラジオ、テレビジョン受信機等に受信障害を与えることがあります。

取扱説明書に従って正しい取り扱いをして下さい。

This equipment is in the Class 1 category (Information Technology Equipment to be used in commercial and/or industrial areas) and conforms to the standards set by the Voluntary Control Council For Interference by Information Technology Equipment aimed at preventing radio interference in commercial and/or industrial areas. Consequently, when used in a residential area or in an adjacent area thereto, radio interference may be caused to radios and TV receivers, etc. Read the instructions for correct handling.

CE NOTICE

Marking by the symbol **CE** indicates compliance of this system to the EMC (Electromagnetic Compatibility) directive of the European Community and compliance to the Low Voltage (Safety) Directive. Such marking is indicative that this system meets or exceeds the following technical standards:

- EN 55022 - "Limits and Methods of Measurement of Radio Interference Characteristics of Information Technology Equipment."
- EN 50082-1 - "Electromagnetic compatibility - Generic immunity standard Part 1: Residential, commercial, and light industry."

SAFETY CERTIFICATIONS

ETL certified to meet Information Technology Equipment safety standards UL 1950 3rd Edition, CSA22.2, No. 950-95, EN 60950 (1992) and IEC 950, 2nd Edition.

CANADIAN IC CS-03 COMPLIANCE STATEMENT

NOTICE: The Industry Canada label identifies certified equipment. This certification means that the equipment meets certain telecommunications network protective, operational and safety requirements. The Industry Canada label does not guarantee the equipment will operate to the user's satisfaction.

Before installing this equipment, users should ensure that it is permissible to be connected to the facilities of the local telecommunications company. The equipment must also be installed using an acceptable method of connection. In some cases, the company's inside wiring associated with a single line individual service may be extended by means of a certified connector assembly (telephone extension cord). The customer should be aware that compliance with the above conditions may not prevent degradation of service in some situations.

Repairs to certified equipment should be made by an authorized Canadian maintenance facility designated by the supplier. Any repairs or alterations made by the user to this equipment, or equipment malfunctions, may give the telecommunications company cause to request the user to disconnect the equipment.

Users should ensure for their own protection that the electrical ground connections of the power utility, telephone lines and internal metallic water pipe system, if present, are connected together. This precaution may be particularly important in rural areas.

Caution: Users should not attempt to make such connections themselves, but should contact the appropriate electric inspection authority, or electrician, as appropriate.

TRADEMARKS

FORE Systems is a registered trademark, and *ForeView* and *PowerHub* are trademarks of FORE Systems, Inc. All other brands or product names are trademarks or registered trademarks of their respective holders.

Table of Contents

List of Figures

List of Tables

Preface

Chapter Summaries	i
Related Documentation	ii
Technical Support	iii
Typographical Styles	iv
Important Information Indicators	iv
Laser Warning	v
Safety Agency Compliance	vi
Safety Precautions	vi
Symbols	vi
Modifications to Equipment	vi
Placement of a FORE Systems Product	vii
Power Cord Connection	vii
Command Syntax	viii

CHAPTER 1 Overview

1.1	Filters	1 - 1
1.1.1	Filter Use	1 - 1
1.1.2	Bridge Filters	1 - 2
1.1.3	Route Filters	1 - 2
1.2	Templates	1 - 3
1.2.1	Bridge Filter Templates	1 - 3
1.2.2	Route Filter Templates	1 - 4
1.3	Rules	1 - 4

CHAPTER 2 AppleTalk

2.1	AppleTalk Filter Basics	2 - 1
2.1.1	Exclusivity	2 - 2
2.1.2	Applying Multiple Filters	2 - 2
2.1.3	Input and Output Filters	2 - 2

Table of Contents

2.2	NBP Forward Filters	2 - 3
2.2.1	Adding Filters	2 - 3
2.2.2	Displaying Filters	2 - 4
2.2.3	Deleting Filters	2 - 6
2.3	Zone Packet Output Filters	2 - 6
2.3.1	Adding Filters	2 - 6
2.3.2	Displaying Filters	2 - 7
2.3.3	Deleting Filters	2 - 8
2.4	Zone Data Input Filters	2 - 8
2.4.1	Adding Filters	2 - 9
2.4.2	Displaying Filters	2 - 10
2.4.3	Deleting Filters	2 - 10
2.5	Zone Data Output Filters	2 - 11
2.5.1	Adding Filters	2 - 11
2.5.2	Displaying Filters	2 - 12
2.5.3	Deleting Filters	2 - 13

CHAPTER 3 Bridge Filters

3.1	Templates	3 - 2
3.1.1	Offset	3 - 2
3.1.2	Mask	3 - 2
3.1.3	Comparator	3 - 2
3.1.4	Offset, Mask, and Comparator Interaction	3 - 3
3.1.4.1	Sending Packets That Evaluate to True	3 - 4
3.2	Working with Templates	3 - 4
3.2.1	Defining Templates	3 - 4
3.2.2	Displaying Templates	3 - 7
3.2.3	Deleting Templates	3 - 8
3.3	Rules	3 - 9
3.3.1	Pre-defined Rules	3 - 9
3.4	Working with Rules	3 - 9
3.4.1	Defining Rules	3 - 10
3.4.2	Displaying Rules	3 - 11
3.4.3	Deleting Rules	3 - 12
3.5	Filters	3 - 12
3.6	Working with Filters	3 - 13
3.6.1	Attaching Rules to Filters	3 - 13
3.6.2	Displaying Filters	3 - 15
3.6.3	Detaching Rules from Filters	3 - 16

CHAPTER 4 Host Filters

4.1	Templates	4 - 1
4.2	Working with Templates	4 - 1
	4.2.1 Defining Templates	4 - 2
	4.2.2 Displaying Templates	4 - 5
	4.2.3 Deleting Templates	4 - 6
4.3	Filters	4 - 6
4.4	Working with Filters	4 - 7
	4.4.1 Defining Filters	4 - 7
	4.4.2 Attaching Filters	4 - 8
	4.4.3 Displaying Filters	4 - 9
	4.4.4 Detaching Filters	4 - 10
	4.4.5 Deleting Filters	4 - 11

CHAPTER 5 IP Filters

5.1	Templates	5 - 2
5.2	Working with Templates	5 - 3
	5.2.1 Defining Templates	5 - 3
	5.2.2 Displaying Templates	5 - 6
	5.2.3 Deleting Templates	5 - 7
5.3	Working with Filters	5 - 8
	5.3.1 Defining Filters	5 - 8
	5.3.2 Displaying Filters	5 - 9
	5.3.3 Deleting Filters	5 - 10
	5.3.4 Attaching Filters to a Segment	5 - 10
	5.3.5 Detaching Filters from a Segment	5 - 11

CHAPTER 6 IP/RIP Import/Export Filters

6.1	Templates	6 - 2
6.2	Working with Templates	6 - 2
	6.2.1 Defining Templates	6 - 3
	6.2.2 Displaying Templates	6 - 4
	6.2.3 Deleting Templates	6 - 5
	6.2.4 Displaying Template Statistics	6 - 6
	6.2.5 Clearing Template Statistics	6 - 6
6.3	Filters	6 - 7
6.4	Working with Filters	6 - 7
	6.4.1 Defining Filters	6 - 7
	6.4.2 Displaying Filters	6 - 8
	6.4.3 Appending Filters	6 - 9
	6.4.4 Inserting Filters	6 - 10

Table of Contents

6.4.5	Undefining Filters	6 - 12
6.4.6	Deleting Filters	6 - 12
CHAPTER 7 IP/OSPF Filters		
7.1	Templates	7 - 1
7.2	Working with Templates	7 - 2
7.2.1	Defining Templates	7 - 3
7.2.2	Displaying Templates	7 - 4
7.2.3	Deleting Templates	7 - 5
7.2.4	Displaying Template Statistics	7 - 5
7.2.5	Clearing Template Statistics	7 - 6
7.3	Filters	7 - 6
7.4	Working with Filters	7 - 7
7.4.1	Defining Filters	7 - 7
7.4.2	Appending Filters	7 - 8
7.4.3	Inserting Filters	7 - 8
7.4.4	Displaying Filters	7 - 10
7.4.5	Undefining Filters	7 - 11
7.4.6	Deleting Filters	7 - 11
CHAPTER 8 IPX RIP/SAP Filters		
8.1	RIP and SAP Filter Types	8 - 1
8.1.1	Exclusivity	8 - 2
8.1.2	Entering IPX RIP and IPX SAP Commands	8 - 2
8.1.3	Types of Control	8 - 3
8.2	IPX RIP Filters	8 - 3
8.2.1	Data Input Filters	8 - 4
8.2.1.1	Adding Filters	8 - 4
8.2.1.2	Displaying Filters	8 - 5
8.2.1.3	Deleting Filters	8 - 6
8.2.2	Data Output Filters	8 - 6
8.2.2.1	Adding Filters	8 - 7
8.2.2.2	Displaying Filters	8 - 8
8.2.2.3	Deleting Filters	8 - 9
8.2.3	Packet Output Filters	8 - 9
8.2.3.1	Adding Filters	8 - 9
8.2.3.2	Displaying Filters	8 - 10
8.2.3.3	Deleting Filters	8 - 11

- 8.3 IPX SAP Filters 8 - 11
 - 8.3.1 Data Input Filters 8 - 12
 - 8.3.1.1 Adding Filters 8 - 12
 - 8.3.1.2 Displaying Filters 8 - 14
 - 8.3.1.3 Deleting Filters 8 - 15
 - 8.3.2 Data Output Filters 8 - 15
 - 8.3.2.1 Adding Filters 8 - 15
 - 8.3.2.2 Displaying Filters 8 - 17
 - 8.3.2.3 Deleting Filters 8 - 17
 - 8.3.3 Packet Output Filters 8 - 18
 - 8.3.3.1 Adding Filters 8 - 18
 - 8.3.3.2 Displaying Filters 8 - 19
 - 8.3.3.3 Deleting Filters 8 - 19

Index

Table of Contents

List of Figures

Figure 3.1 Use of Offset, Mask, and Comparator 3 - 3
Figure 3.2 TCP/IP Packet Template 3 - 4

List of Figures

List of Tables

Table 3.1	Template Definitions	3 - 6
Table 3.2	Rule Definition Examples	3 - 10
Table 8.1	Server Types.	8 - 13

List of Tables

Preface

The intent of this manual is to supply users of the *ForeRunner* ASN-9000 with all the necessary information to setup and configure filters in the ASN-9000 successfully. If questions or problems with the installation arise, please contact FORE Systems' Technical Support.

Chapter Summaries

Chapter 1 - Overview - Provides an overview of filters and how they are used in the ASN-9000. General information on filters, templates and rules is discussed.

Chapter 2 - AppleTalk - Describes AppleTalk filters and templates and how to attach them to segments. Each of the AppleTalk filter and template command are discussed and examples are provided on defining and attaching.

Chapter 3 - Bridge Filters - Describes bridge filters, templates and rules and how to define and attach a bridge filter to MAC address nodes or segments. All of the bridge template, rule and filter commands are discussed and examples are provided on defining and applying templates and rules to bridge filters.

Chapter 4 - Host Filters - Describes host templates and filters. Each of the Host subsystem template and filter command options are discussed. Examples are provided to assist in defining host templates and filters.

Chapter 5 - IP Filters - Describes IP filters and templates. All of the IP template and filter commands are discussed and examples are provided to assist in defining IP templates and filters.

Chapter 6 - IP/RIP Import/Export Filters - Describes IP/RIP import/export filters and templates. The IP/RIP subsystem commands to define IP/RIP import and export filters and templates are discussed. Examples are provided to assist in creating and applying IP/RIP import and Export filters.

Chapter 7 - IP/OSPF Filters - Describes the IP/OSPF filter and template commands. Examples are provided to assist in defining and applying IP/OSPF filters and templates.

Chapter 8 - IPX RIP/SAP Filters - Describes the IPX RIP/SAP filter and template commands. Examples are provided to assist in defining templates and filters for IP/OSPF interfaces.

Related Documentation

- *ForeRunner ASN-9000 Installation and Maintenance Manual*, MANU0166-02, June 1, 1998
- *ForeRunner ASN-9000 Software Reference Manual*, MANU0167-02, June 1, 1998
- *ForeRunner ASN-9000 Protocols Reference Manual*, MANU0271-02, June 1, 1998
- *ForeRunner ASN-9000 Release Notes*, MANU 0274-03, June 1, 1998.

Technical Support

In the U.S.A., customers can reach FORE Systems' Technical Assistance Center (TAC) using any one of the following methods:

1. Select the "Support" link from FORE's World Wide Web page:

<http://www.fore.com/>

2. Send questions, via e-mail, to:

support@fore.com

3. Telephone questions to "support" at:

800-671-FORE (3673) or 724-742-6999

4. FAX questions to "support" at:

724-742-7900

Technical support for customers outside the United States should be handled through the local distributor or via telephone at the following number:

+1 724-742-6999

No matter which method is used to reach FORE Support, customers should be ready to provide the following:

- A support contract ID number
- The serial number of each product in question
- All relevant information describing the problem or question

Typographical Styles

Throughout this manual, specific commands to be entered by the user appear on a separate line in bold typeface. In addition, use of the Enter or Return key is represented as <ENTER>. The following example demonstrates this convention:

```
cd /usr <ENTER>
```

Commands or file names that appear within the text of this manual are represented in the following style: "...the fore_install program will install this distribution"

Important Information Indicators

To call your attention to safety and otherwise important information that must be reviewed to insure correct and complete installation, as well as to avoid damage your system, FORE Systems utilizes the following *WARNING/CAUTION/NOTE* indicators.

WARNING statements contain information that is critical to the safety of the operator and/or the system. Do not proceed beyond a **WARNING** statement until the indicated conditions are fully understood or met. This information could prevent serious damage to the operator, the system, or currently loaded software, and will be indicated as:

WARNING!



Hazardous voltages are present. To lessen the risk of electrical shock and danger to personal health, follow the instructions carefully.

Information contained in **CAUTION** statements is important for proper installation/operation. **CAUTION** statements can prevent possible equipment damage and/or loss of data and will be indicated as:

CAUTION



You risk damaging your equipment and/or software if you do not follow these instructions.

Information contained in NOTE statements has been found important enough to be called to the special attention of the operator and will be set off from the text as follows:



Steps 1, 3, and 5 are similar to the installation for the computer type above. Review the previous installation procedure before installation in your particular model.

Laser Warning

**Class 1 Laser Product:
This product conforms to
applicable requirements of
21 CFR 1040 at the date of
manufacture.**

Class 1 lasers are defined as products which do not permit human access to laser radiation in excess of the accessible limits for Class 1 for applicable wavelengths and durations. These lasers are safe under reasonably foreseeable conditions of operation.

The *CellPath* 300 OC-3c/STM1 single-mode physical layer modules contain Class 1 lasers.

Safety Agency Compliance

This preface provides safety precautions to follow when installing a FORE Systems, Inc., product.

Safety Precautions

For your protection, observe the following safety precautions when setting up your equipment:

- Follow all warnings and instructions marked on the equipment.
- Ensure that the voltage and frequency of your power source matches the voltage and frequency inscribed on the equipment's electrical rating label.
- Never push objects of any kind through openings in the equipment. Dangerous voltages may be present. Conductive foreign objects could produce a short circuit that could cause fire, electric shock, or damage to your equipment.

Symbols

The following symbols appear in this book.

WARNING!



Hazardous voltages are present. If the instructions are not heeded, there is a risk of electrical shock and danger to personal health.

CAUTION



If instructions are not followed, there is a risk of damage to the equipment.

Modifications to Equipment

Do not make mechanical or electrical modifications to the equipment. FORE Systems, Inc., is not responsible for regulatory compliance of a modified FORE product.

Placement of a FORE Systems Product

CAUTION



To ensure reliable operation of your FORE Systems product and to protect it from overheating, openings in the equipment must not be blocked or covered. A FORE Systems product should never be placed near a radiator or heat register.

Power Cord Connection

WARNING!



FORE Systems products are designed to work with single-phase power systems having a grounded neutral conductor. To reduce the risk of electrical shock, do not plug FORE Systems products into any other type of power system. Contact your facilities manager or a qualified electrician if you are not sure what type of power is supplied to your building.

WARNING!



Your FORE Systems product is shipped with a grounding type (3-wire) power cord. To reduce the risk of electric shock, always plug the cord into a grounded power outlet.

Command Syntax

The following expressions are used in this manual when describing command syntax:

AaBbCcDd A term that is being defined. Example:

IP Helper is an enhancement to the `ip` subsystem that allows a system to be boot from a server separated from the boot client by a gateway.

AaBbCcDd A command name. Commands are case-sensitive; they should always be issued in lowercase. Example:

`dir`

| 1) Separates the full and terse forms of a command or argument:

- The full form is shown on the left of the |.
- The terse form is shown on the right of the |.

Example:

`dir | ls`

When the command or argument is entered, either the full form or terse form may be used. In this example, either `dir` or `ls` can be used.

2) Separates mutually exclusive command arguments. Example:

`active-ama|aa cset p[primary]|b[ackup] <slot>|all`

In this example, the command `active-ama|aa` can accept either `active-ama` or `aa`, but not both.

[] Enclose optional command arguments or options. Example:

`active-ama|aa [show] [linemode|lm <slot>]|all`

In this example, the [] enclose an optional argument. The command can be issued without the argument(s) shown in []. However, if specified, the argument must be one of the two options listed between the [].

<AaBbCcDd> Indicates a parameter for which a value is supplied by the operator. When used in command syntax, *<italics>* indicates the value to be supplied. Example:

savecfg <filename>

In this example, *<filename>* is a parameter for which a value must be supplied with the command is issued.

AaBbCcDd Indicates a field name or a file name.

An example of a field name is when booting the software, the `login:` prompt is displayed.

A filename example is when booting the software, the system looks for a file name `cfg`.

Indicates text (commands) displayed by the software or typed at the command prompt. To distinguish typed input from command output, the typed input is shown in bold typeface. Example:

```
16:ASN-9000:system# bootinfo
Tue Jan 20 15:46:25 1998 start
Tue Jan 20 15:46:34 1998 nvram boot order: fm
boot device: m
17:ASN-9000:system#
```

In this example, the user enters **bootinfo** and the software responds with:

```
Tue Jan 20 15:46:25 1998 start
Tue Jan 20 15:46:34 1998 nvram boot order: fm
boot device: m
```

Preface

The *ForeRunner* ASN-9000 supports the use of filters to selectively grant or prohibit access to areas of the network. Different types of filters can be configured to control different aspects of the network. Filters can be configured for the following protocols:

- AppleTalk (refer to *Chapter 2, AppleTalk*)
- Bridge (refer to *Chapter 3, Bridge Filters*)
- Host (TCP) (refer to *Chapter 4, Host Filters*)
- IP (refer to *Chapter 5, IP Filters*)
- IP/RIP (refer to *Chapter 6, IP/RIP Import/Export Filters*)
- IP/OSPF (refer to *Chapter 7, IP/OSPF Filters*)
- IPX RIP/SAP (refer to *Chapter 8, IPX RIP/SAP Filters*)

1.1 Filters

Filters control how packets are handled within a network. Access to portions of the network can be controlled in different ways depending on the type of traffic being handled. The ASN-9000 supports two basic types of filters:

- Bridge filters
- Route filters

1.1.1 Filter Use

The protocols these filters use to control packet access in the following ways:

- | | |
|---------------------------|--|
| Data Input Filter | Operates on the receipt of packet traffic (typically a receiving segment or network interface). When the ASN-9000 receives a packet, data input filters accept or reject information in the received packet. |
| Data Output Filter | Operates on the transmission of packet traffic (typically a transmitting segment or network interface). Before the ASN-9000 transmits a packet, data output filters control whether specific entries are included in, or omitted from, the packet. |

Packet Output Filter Operates on the transmission of packet traffic (typically a transmitting segment or network interface). Before the ASN-9000 transmits a packet, packet output filters control whether the entire packet is transmitted or not.

1.1.2 Bridge Filters

Bridge packets are controlled by the segment numbers on which packets are sent and/or received. The ASN-9000 controls packets as they enter or exit the port to which the segment is connected. For more information on bridge filters, refer to *Chapter 3, Bridge Filters*.

1.1.3 Route Filters

Route filters control routed traffic. The way packets are controlled is very similar among routed traffic, yet the syntax and specifics of each type of router filter depends on the protocol. If the traffic is being received and transmitted, the protocol-specific packets can be controlled by filters configured on segments. Some route filters, like bridge filters, incorporate the use of templates. Route filters can take two forms:

- Packet filters
- Route filters

Packet filters pass or block entire packets based on user-defined criteria and are often addressed to the ASN-9000. Packets are passed or blocked based on the routes advertised in the packets. Route filters often are destined for a device other than the ASN-9000. These filters apply to packets that carry route information, such as protocol-specific updates or report packets.

Those protocols that do not use templates, use filters to directly compare packets and control the packet's access to the ASN-9000 and the rest of the network. The following protocols use filters to directly control packet access:

- IPX RIP
- IPX SAP
- AppleTalk

1.2 Templates

Templates are user-defined structures that compare portions, or all, of a packet based on user-defined conditions. Templates can be applied to Bridge, IP, Host and OSPF filters.

Because bridge and route filters operate on different layers, using different types of data to transmit and receive packets, bridge and route filters use different conditions to comprise the templates. However, regardless of the differences, templates define a set of conditions that are compared to packets.

1.2.1 Bridge Filter Templates

Bridge filters, like most filters, require templates. Templates are a set of conditions used to filter bridge packets. The ASN-9000 filters packets by matching them with the conditions specified in the template. If the bytes in the packet match the template's pattern, the result is true. Packets that evaluate to true are forwarded or blocked depending on the template definition. Bridge filters use three unique elements in the creation of templates:

offset	Tells where to begin comparing the packet to the template.
mask	Indicates how much of the packet to ignore when filtering.
comparator	Indicates how much of the packet is compared to the template and the value to which the packet should be compared.

For more information on bridge templates, refer to *Chapter 3, Bridge Filters*.

1.2.2 Route Filter Templates

Some, but not all, route filters require templates. Templates are a set of conditions used to compare against a routed packet. Packets are filtered by matching them with the conditions specified in the template. If the bytes in the packet match the conditions specified in the template, the result is true. Packets that evaluate to true are forwarded or blocked depending on the template definition. The following protocols use templates in their respective route filters:

- IP
- IP/RIP
- IP/OSPF
- Host (TCP)

1.3 Rules

Rules can be applied to packets and segments when using bridge filters. Rules are a combination of templates and logical operators (such as AND (&), OR (|) and NOT (~)). Logical operators can be used to specify the way that multiple templates interact when comparing packets to a user defined set of conditions.

In the ASN-9000, rules are not defined using a specific command. However, the concept of filter rules is included in the process used in defining filter rules. Rules can be used to assign a combination of templates to a target interface or segment.

AppleTalk filters provide a way to ensure security for AppleTalk networks configured on a ASN-9000. AppleTalk is similar to other types of route protocols in that it filters packets that are routed, not bridged.

AppleTalk networks use a network address known as a “network range”. In concept, the network range is similar to an IP address and consists of a network and node ID. AppleTalk filters can filter packets based on the network range.

Zones are another element of an AppleTalk network that can also be configured on AppleTalk network ranges. By specifying a zone, a name is associated with a logical grouping of nodes on a particular part of an AppleTalk network.

2.1 AppleTalk Filter Basics

Different AppleTalk filters perform filtering based on different criteria. Some packets are filtered by network range, some specific types of AppleTalk packets are filtered by the zone names contained within them, and some packets are filtered by the segment on which they are sent and received. The ASN-9000 implementation of AppleTalk filters feature the following types of filters:

- Name-Binder-Protocol (NBP) Forward filters
- Zone Packet- Output filters
- Zone Data-Input filters
- Zone Data Output filters

AppleTalk filtering results in two outcomes:

- pass (allow certain traffic to proceed on the AppleTalk network)
- block (prevent certain traffic from proceeding on the AppleTalk network)

For more information about AppleTalk commands, refer to the *ForeRunner ASN-9000 Protocols Reference Manual*.

2.1.1 Exclusivity

AppleTalk filters of the same type (NBP-forwarding filters, zone packet-output filters, zone data-input filters, and zone data-output filters) are mutually exclusive. If a filter is defined that explicitly receives or sends specific information, all other information is implicitly discarded. For example, if a zone data input filter is defined that explicitly accepts updates from a specific network, all other updates from that network are discarded. To accept additional updates from that network, additional filters need to be defined. However, updates received from other networks are not affected.

If secure access to just a few networks is needed, it is generally easier to define filters that block or discard update information sent on or received from just those networks. All update information not explicitly blocked is forwarded. However, if tight security is required, define filters that explicitly allow only specific updates to be sent or received.

2.1.2 Applying Multiple Filters

Filters are applied in ascending numerical order (from the lowest filter number to the next highest). Therefore, filters should be defined in a “most important” to “least important” order. If more than one filter is defined, the following rules determine how the filters are applied:

The filtering process accepts or discards packets when a filter finds the AppleTalk zone or network number that it is constructed to match. If a match is made, the filter performs the defined function. AppleTalk zone-packet output, data-input and output, and NBP filters work in the following manner:

- If all filters are `pass` or `block` and there is no match, the zone or NBP object is hidden or blocked.
- If all filters are `pass` or `block` and there is no match, the zone or NBP object is reported.
- If both `report` and `hide` filters, or `send` and `block` filters, are defined and there is no match, the zone is hidden. To change this behavior, define the last filter (filter number 128) as a report filter that matches all zones.

2.1.3 Input and Output Filters

When an AppleTalk zone filter is defined, the network range and segment number or the zone name and segment number are supplied. For NBP filters, the zone name and segment number or the AppleTalk object type and network range are supplied. The use of these arguments depends on whether an input or output filter is being defined:

- Input filter** Operates on the receiving end of the report or update. When a network or a specific segment receives a report or update, input filters accept or reject information in the update. Zone accept filters are input filters.
- Output filter** Operates on the sending end of the report or update. Before an update or report is sent for an AppleTalk network on a specific segment, output filters report or discard entries in the update or report. Zone-update filters, zone-report filters, and NBP filters are output filters.



AppleTalk filters do not require templates. When the filter is assigned, the conditions specified are matched against packets, the actions that the filter takes when packets are matched, and the segment on which the filter is applied.

2.2 NBP Forward Filters

AppleTalk Name-Binder-Protocol (NBP) links an AppleTalk device to a network socket, NBP forward filters enable the ASN-9000 to respond to or forward an NBP Lookup request for a network device. AppleTalk devices can be located by name, type, or zone. With NBP forward filters, the ASN-9000 can control the NBP Lookup request being forwarded to a zone on a particular segment.

2.2.1 Adding Filters

The `nbp-fwd-filter add` command is used to create NBP filters. This command enables the filter conditions to be specified, assigns a filter action, and associates the filter with a segment. When the filter has been created, the ASN-9000 controls whether or not an NBP Lookup packet for a particular zone is forwarded on a particular segment. The syntax of the `nbp-fwd-filter |nff add` command is:

```
nbp-fwd-filter|nff add <filnum> b[lock]|p[ass] <seg> <zone>
```

where

<filnum> Specifies the filter number. Valid filter numbers are 1 through 128. The filter number is analogous to a name and is used to reference the filter. Filter numbers do not imply the order of filtering.

b[lock]|p[ass] Specifies the action the filter is to perform when packets are found that match the conditions specified. If **block** is specified, then the matching NBP Lookup packets are discarded. If **pass** is specified, then the matching NBP Lookup packets are forwarded.

<seg> Specifies the segment on which the PowerHub should block or pass the NBP Lookup packets that match the conditions specified.

The syntax requires that the slot containing the segment, as well as the segment itself, be specified. For example, this “slot.segment” format for segment 6 in card slot 1 is expressed as: **1.6**

<zone> Specifies the zone to be filtered. Packets analyzed on the specified zone are either forwarded (**pass**) or discarded (**block**). The specified zone does not need to exist before configuring a filter.

In the following example, filter number 12 is configured to block NBP Lookup packets from reaching devices in the zone “sales” located on segment 1.4.

```
240:ASN9000:atalk# nbp-fwd-filter add 12 block 1.4 sales
nbp-fwd-filter add 12 block 1.4 sales
Ok
242:ASN9000:atalk#
```

2.2.2 Displaying Filters

The **nbp-fwd-filter show** command is used to display defined NBP filters. This command displays the following information about the defined NBP filters:

- filter number
- filter’s action
- segment number on which the filtering process occurs
- zone names that are filtered

The syntax of the this command is:

```
nbp-fwd-filter | nff [show] [<filnum>[,<filnum>...]]
```

where

- [<filnum>]** Specifies the filter to display. Valid filter numbers are 1 through 128. If no filters are specified, all defined NBP filters are displayed.
- [,<filnum>]** Specifies additional filters to display. If more than one filter is specified, separate each with a comma.

In the following example, information on the currently configured filters. As shown below, filter 45 blocks NBP Lookup packets from proceeding into “shipping” on segment 2, filter 101 blocks NBP Lookup packets from proceeding to “sales” on segment 4, and filter 111 allows NBP Lookup packets to pass to the “marketing” zone on segment 6.

```
251:ASN9000:atalk# nff
NBP forward filters:
Fil  Action  Segment  Zone-name
---  -
45   block    2        shipping
101  block    4        sales
111  pass     6        marketing
245:ASN9000:atalk#
```



AppleTalk filters are presented in numerical order. Notice that the “slot.segment” format for entering segment numbers, displays only the segment portion of the <slot.segment> specified in the **add** command.

AppleTalk zones can be configured with spaces before and after zone names, and between zone names if the zone name uses multiple words (for example, “customer support.”) For more information about AppleTalk zone names, refer to the *ForeRunner ASN-9000 Protocols Reference Manual*.

2.2.3 Deleting Filters

The `nbp-fwd-filter delete` command is used to delete an NBP filters. This command deletes one, or all NBP filters. The syntax of this command is:

```
nbp-fwd-filter|nff delete <filnum>|all
```

where

<filnum>|all Specifies the NBP filters to delete. Valid filter numbers are 1 through 128. If **all** is specified, then all defined NFF filters are deleted.

In the following example, NBP filter 111 is deleted.

```
9:ASN9000:atalk# nff delete 111
Ok
10:ASN9000:atalk#
```

2.3 Zone Packet Output Filters

Zone Information Protocol (ZIP) packets use used to inform routers on an AppleTalk network about zones that are configured on an AppleTalk Network, and on what network range each zone is configured. ZIP packets are sent through an AppleTalk network intermittently to refresh each AppleTalk router's zone table with the network to zone name information.

Zone packet output filters control zone data packets that are being transmitted from the ASN-9000. Unlike other AppleTalk filters, they do not filter out specific bytes of data from a packet. Instead, these filters allow or disallow the entire zone data packet from traversing a segment.

2.3.1 Adding Filters

The `zone-pkt-output-filter|zpod add` command is used to add zone packet output filters. This command can be used to specify to filter out any zone data packets transmitted on a particular segment in a specified network range. The syntax of this command is:

```
zone-pkt-output-filter|zpod add <filnum> b[lock]|p[ass] <seg>
<netrange>
```

where

<filnum> Specifies the filter number. Valid filter numbers are 1 through 128.

b[lock] p[ass]	Specifies the action the filter is to take when a packet that matches the conditions is encountered. If block is specified, then packets that match are discarded. If pass is specified, then packets that match are forwarded to their destinations (or to other filters, if defined).
<seg>	Specifies the segment on which zone data packets are forwarded or discarded. Specify only one segment at a time.
<netrange>	Specifies the network range that contains the segment noted in <seg> . Zone data packets are controlled on only the segment specified in <seg> . Filtering does not occur on any other segment in the network range.

In the following example, filter 110 being configured to permit packets to be transmitted on segment 1.1 of AppleTalk network range 101-110.

```
13:ASN9000:atalk# zpof add 110 pass 1.1 101-110
Ok
14:ASN9000:atalk#
```

2.3.2 Displaying Filters

The **zone-pkt-output-filter show** command is used to display the defined zone packet filters. This command displays the following information:

- filter number
- filter's action
- network range on which the filtering process occurs
- zone names that are filtered

The syntax of this command is:

```
zone-pkt-output-filter | zpof [show] [<filnum>[,<filnum>...]]
```

where

<filnum>	Specifies which filter to display. Valid filter numbers are 1 through 128. If no filter number is specified, then all filters are displayed.
[,<filnum>...]	Specifies any additional filters to display. If more than one filter is specified, separate each with a comma.

In the following example, the defined zone data output filters are displayed. The display shows the number assigned to the filter, the action that the filter takes when it finds packets that match the conditions specified, the segment and network range combination where the ZIP packets are to be forwarded or discarded from.

```
16:ASN9000:atalk# zpod
ZIP Packet Output Filters:
Fil  Action  Segment  Range
---  -
110  pass     1.2      101-110
17:ASN9000:atalk#
```

2.3.3 Deleting Filters

The `zone-pkt-output-filter delete` command is used to delete zone packet output filters. This command removes the filter from the available filters. To modify a zone output packet filter, it is necessary to first delete the filter and then to redefine it. The syntax of this command is:

```
zone-pkt-output-filter | zpod delete <filnum> | all
```

where

<filnum>|all Specifies the filter number to delete. Valid filter numbers are 1 through 128. If **all** is specified, then all zone packet output filters are deleted.

In the following example, zone packet output filter 110 is deleted from the zone packet output filter definitions.

```
19:ASN9000:atalk# zpod delete 110
Ok
20:ASN9000:atalk#
```

2.4 Zone Data Input Filters

Zone data input filters allow or deny ZIP packets to be received from a specified AppleTalk range. Filtering occurs by zone names in the ZIP packets, and the ASN-9000 looks for these on the specified range. When a ZIP packet is received, the packet is examined. The packets that contain the specified zone name are passed, or blocked, depending on the filter.

2.4.1 Adding Filters

The `zone-data-input-filter add` command is used to create a zone data-input filter. When creating the filter, the conditions that are to be matched against ZIP packets, the action the filter is to take when it finds a match, and the network range and segment on which ZIP packets are controlled is specified. The syntax of this command is:

```
zone-data-input-filter | zdif add <filnum> b[lock] | p[ass]
                        <netrange> | all <zone> | *
```

where

<filnum>	Specifies the filter number. Valid filter numbers are 1 through 128.
b[lock] p[ass]	Specifies the action the filter must take when ZIP packets that match the conditions set are encountered. If block is specified, then the matched packets are discarded. If pass is specified, then the matched packets are passed on to the next router in the network.
<netrange> all	Specifies the network range on which to filter packets that match the conditions set. If all is specified, then ZIP packets received from all network ranges are filtered, provided the ZIP packets contain the specified zone(s).
<zone> *	Specifies the zone(s) to filter. Specify one zone per filter. If * is specified, then all zones in the ZIP are filtered.

In the following example, zone data-input filter 11 is defined. This filter blocks the receipt of ZIP packets from AppleTalk network range 114-115 on all zones.

```
26:ASN9000:atalk# zdif add 11 block 114-115 *
Ok
27:ASN9000:atalk#
```

2.4.2 Displaying Filters

The `zone-data-input-filter show` command is used to display the defined zone data input filters. This command displays:

- filter number
- action the filter must take
- network range on which the filtering process occurs
- zone names that are filtered

The syntax of this command is:

```
zone-data-input-filter|zdif [show] [<filnum>[,<filnum>...]]
```

where

[<filnum>] Specifies which filter to display. Valid filter numbers are 1 through 128. If no filters are specified, then all filters are displayed.

[,<filnum>...] Specifies additional filters to display. If more than one filter is specified, separate each with a comma.

The following example displays the defined zone data-input filters. No specific filters are stated in the command, so all filters are displayed. Filter 11 is defined to block all zone information on network range 114-115 and filter 13 is defined to pass zone information received for “manufacturing” on network range 115-117.

```
30:ASN9000:atalk# zdif
ZIP Data Input Filters:
Fil Action  Range      Zone-Name
-----
11  block   114-115    *
13  pass    115-117    manufacturing
31:ASN9000:atalk#
```

2.4.3 Deleting Filters

The `zone-data-input-filter delete` command is used to delete zone data input filters. To modify a filter it is necessary to delete the filter and then add a new filter. The syntax of this command is:

```
zone-data-input-filter|zdif delete <filnum>
```

where

<filnum> Specifies the zone data input filter to delete. Valid filter numbers are 1 through 128. All zone data input filters can be deleted by specifying **all** instead of a specific filter number.

In the following example zone data input filter 11 is deleted.

```
33:ASN9000:atalk# zdif delete 11
Ok
34:ASN9000:atalk#
```

2.5 Zone Data Output Filters

Zone data output filters allow or deny ZIP reports to be sent on the specified network range and segment combination. The filtering criterion is the zone names in the ZIP reports, and the ASN-9000 looks for ZIP reports on the specified range. The packets that contain the specified zone are passed or blocked depending on how the filter is set up.

2.5.1 Adding Filters

The **zone-data-output-filter add** command is used to create zone data-output filters. To add a filter, assign the filter a number, specify whether it is a block or pass filter, define the network range to which access is controlled, and specify which zone names should be matched during filtering. The syntax of this command is:

```
zone-data-output-filter|zdof add <filnum> b[lock]|p[ass]
<netrange>|all <zone>|*
```

where

<filnum> Specifies the filter number. Valid filter numbers are 1 through 128.

b[lock]|p[ass] Specifies the action the filter is to take when packets matching the specified conditions are encountered. If **block** is specified, then the matching packets are discarded. If **pass** is specified, then the matching packets are passed on to the rest of the network.

- <netrange>|all** Specifies the network range on which to filter packets. If **all** is specified, then packets received from all network ranges are filtered, provided the packets contain the specified zone(s).
- <zone>|*** Specifies the zone that must be filtered. Specify one zone per filter or specify all zones. If ***** is specified, all zones are filtered.

The following example adds a zone data output filter to block ZIP packets from being sent to “sales” on network range 119-121.

```
36:ASN9000:atalk# zdof add 12 block 119-121 sales
Ok
37:ASN9000:atalk#
```

2.5.2 Displaying Filters

The **zone-data-output-filter show** command is used to display the defined zone data-output filters. This command displays:

- filter number
- action the filter is to take
- network range on which the filtering process occurs
- zone names that are filtered

The syntax of this command is:

```
zone-data-output-filter|zdof [show] [<filnum>[,<filnum>...]]
```

where

- <filnum>** Specifies which filter to display. Valid filter numbers are 1 through 128. If no filters are specified, then all are displayed.
- ,<filnum>** Specifies any additional filters to display. If more than one filter is specified, separate each with a comma.

The following example displays all defined zone data outputfilters. In the display, filter 17 passes packets to “engineering” on network range 114-115 and filter 19 blocks packets for “marketing” on network range 117-124.

```
40:ASN9000:atalk# zdof
ZIP Data Output Filters:
Fil Action  Range          Zone-Name
-----
17  pass    114-115    engineering
19  block   117-124    marketing
41:ASN9000:atalk#
```

2.5.3 Deleting Filters

The `zone-data-output-filter delete` command is used to delete zone data-output filters. To modify a zone data-input filter, delete the filter first, then add a new filter. The syntax of this command is:

```
zone-data-output-filter|zdof delete <filnum>|all
```

where

<filnum>|all Specifies which filter to delete. Valid filter numbers are 1 through 128. If **all** is specified, then all zone data output filters are deleted.

The following example deletes filter 19.

```
52:ASN9000:atalk# zdof del 19
Ok
53:ASN9000:atalk#
```

AppleTalk

CHAPTER 3

Bridge Filters

The *ForeRunner* ASN-9000 supports Bridge filters to allow or disallow bridged packets sent to or received from specified segments or MAC addresses. Bridge filtering augments the standard bridging algorithms used in the bridging engine. By defining templates and rules, and associating them with specific segments, further control over which packets are sent or received is gained.

Before forwarding a packet, the packet is checked against user-defined templates and rules. In general, a rule returns a value of `true` or `false` by evaluating a combination of templates that compare a pattern against a specified portion of the packet. If a rule returns a value of `true`, the conditions specified in the filter are applied to the packet. If all rules return a value of `false`, the packet is not filtered.

CAUTION



Bridge filters affect packets that are bridged between segments in either a pure bridging or a virtual-LAN configuration. Bridge filters are not applied to packets that are routed between segments, or generated by or addressed to the ASN-9000 itself. Bridge filters are applied to broadcast packets in a pure bridging or virtual-LAN configuration.

Bridge filters are comprised of:

- Templates (*Section 3.1*)
- Rules *Section 3.3*)
- Filters (*Section 3.5*)

Bridge filter commands allow the user to:

- Define/Display/Delete templates
- Define/Display/Delete rules
- Attach/Detach filters
- Display where filters are attached

The filter related commands are only a part of the bridge subsystem commands. For a complete listing of commands in the bridge subsystem, refer to the *ForeRunner ASN-9000 Software Reference Manual*.

3.1 Templates

Templates are user-defined structures applied to packets. Templates can return a value of `true` or `false`. Up to 98 templates can be defined. An additional template, template 99, is pre-defined and cannot be altered. Template 99 is a “match anything” template that is described in *Section 3.1.4.1*. Templates have the following three components:

- offset
- mask
- comparator

3.1.1 Offset

An offset is a pointer that specifies the displacement, in octets, from the beginning of the packet. The offset always begins at the start of the packet (bypassing the preamble). The offset is normally a multiple of 4 in the range 0 through 112 decimal. Four octets, at displacement offset through offset+3 from the beginning of the packet, are checked.

3.1.2 Mask

A mask is a four-byte (32-bit) number normally specified as eight hexadecimal digits. The bytes in the mask are numbered from 0 to 3, starting with the high-order byte (“big-endian” format). Each byte (i) of the mask is ANDed with the octet in the packet at displacement (offset+i) to form a 4-byte masked value.

3.1.3 Comparator

A comparator is a 4-byte number normally specified as eight hexadecimal digits. If the masked value equals the comparator, then the template returns a value of `true`. If it does not, it returns a value of `false`. When the comparison is finished, the result of the comparison depends on the action specified in the filter.

3.1.4 Offset, Mask, and Comparator Interaction

Figure 3.1 illustrates how the offset, mask, and comparator are used during the filtering process.

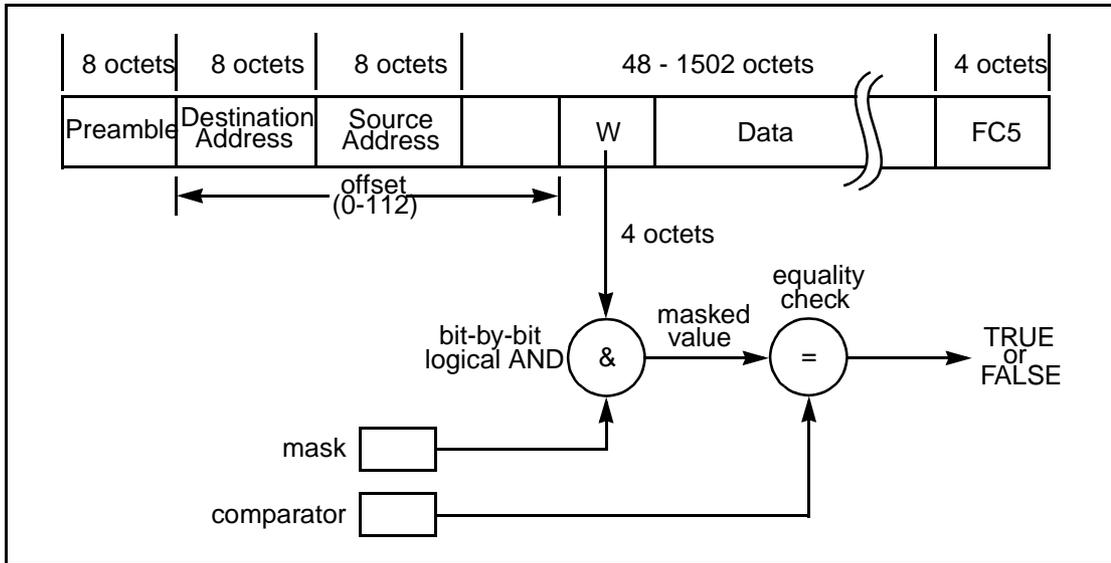


Figure 3.1 - Use of Offset, Mask, and Comparator

For example, suppose the Ethernet packet `type` field for a particular value is to be checked. As shown in Figure 3.2, the `type` field consists of two octets beginning at offset 12. A value of 0800 hex in this field indicates a TCP/IP packet. As shown in the figure, a template that returns `true` only for TCP/IP packets has an offset of 12, a mask of FFFF0000, and a comparator of 08000000.

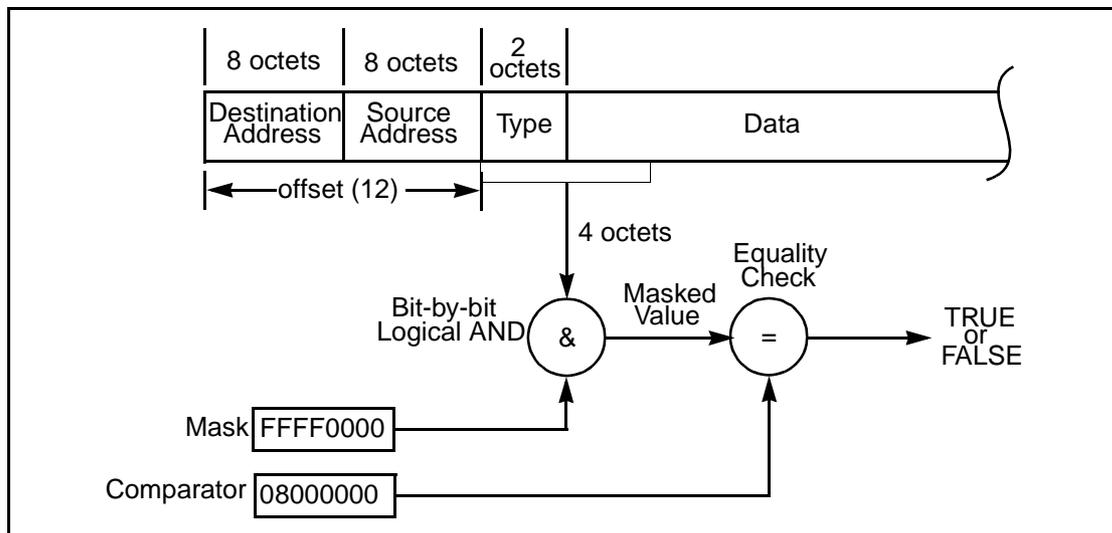


Figure 3.2 - TCP/IP Packet Template

3.1.4.1 Sending Packets That Evaluate to True

The template definition does not automatically send a packet that has evaluated to `true` through the entire series of template-to-packet comparisons. In order for such a packet to proceed through the network, the packet must match one last template, a “match anything” template. This template, template 99, is included so that packets that have evaluated to `true` all the way through the filtering process can be sent to the rest of the network. Template 99 can be seen if the `config templates` command is issued. Template 99 cannot be deleted or modified in any way.

3.2 Working with Templates

When creating templates, they are compared against the different values specified. The matching occurs at certain locations within the packet, as described in *Section 3.1.4*. The following sections explain how to create and use bridge templates.

3.2.1 Defining Templates

Use the `template define` command to define a template. Templates are the first step in defining a bridge filter. The syntax is as follows:

```
template define <tnum> [size=w|h|b] off=<num> mask=<mask> comp=<comp>
```


NOTE

When command objects are specified, the equal sign (=) must be included. If the equal sign is not included, an error message is returned.

where

<tnum>	Specifies the template number. Valid template numbers are 1 through 98. Template numbers imply no logical order for filtering.
[size=w h b]	Optional parameter that specifies the size of the filter. This argument indicates that the mask and comparator are specified as whole word (4-byte), half-word (2-byte), or single-byte quantities. When using h , the offset is an even number in the range 0–126. When using b , the offset is any number in the range 0–127.
off=<num>	Specifies, in decimal format, the number of bytes bypassed (starting with the first byte after the preamble) before the template is applied during filtering. For example, if the offset is 8, then the first 8 bytes are bypassed. Filtering would start on the ninth byte.
mask=<mask>	Specifies the amount of the packet to be compared with the template. In effect, the mask hides the part of the bridge packet that is not to be filtered. The offset value is a 2- to 8-digit hexadecimal number in the range 00000000–ffffffff.
comp=<comp>	Specifies the part of the packet compared to the template. The comparator value is a 2- to 8-digit hexadecimal number in the range 00000000–ffffffff.

Table 3.1 provides some examples of template definitions. For most template definitions, only a 2- or 1-byte field is pertinent, but a full 4-byte mask and comparator are defined. Since the offset must be a multiple of 4 bytes, the bytes of interest must be specified in the appropriate position within an aligned 4-byte value. The examples in Table 3.1 are presented twice. The first fourteen templates are defined without using the optional `[size=w|h|b]` argument. The second group of fourteen uses the optional argument. This shows that bridge templates can be defined with or without the use of the optional argument.

Table 3.1 - Template Definitions

Command	Comment
template define 1 0 FFFFFFF0 FFFFFFF0	Select broadcast packets
template define 2 0 01000000 01000000	Select broadcast or multicast packets
template define 3 12 FFFF0000 08000000	Select TCP/IP packet type
template define 4 24 0000FFFF 00005500	Class A IP source network 55 ₁₆ (85 ₁₀)
template define 5 28 0000FFFF 00005500	Class A IP destination network 55 ₁₆ (85 ₁₀)
template define 6 32 0000FFFF 00003A3A	Select TCP source segment 3A3A
template define 7 12 FFFF0000 08060000	Select ARP-request packet-type
template define 8 28 0000FFFF 00000464	Select 3Com name server socket
template define 9 12 FFFF0000 06000000	Select XNS packet type
template define 10 12 FFFF0000 809B0000	Select Kinetics EtherTalk
template define 11 16 FF000000 15000000	Select AppleTalk source address 15
template define 12 16 00FF0000 00220000	Select AppleTalk destination address 22
template define 13 12 FFFF0000 60040000	Select DEC LAT packet type
template define 14 12 FFFF0000 60060000	Select DEC DECnet packet type
template define 1 w 0 FFFFFFF0 FFFFFFF0	Select broadcast packets
template define 2 b 0 01 01	Select broadcast or multicast packets
template define 3 h 12 FFFF 0800	Select IP packet type
template define 4 h 26 FFFF 5500	Class A IP source network 55 ₁₆ (85 ₁₀)
template define 5 h 30 FFFF 5500	Class A IP destination network 55 ₁₆ (85 ₁₀)
template define 6 h 34 FFFF 3A3A	Select TCP source segment 3A3A ₁₆ (238496 ₁₀)
template define 7 h 12 FFFF 0806	Select ARP-request packet type
template define 8 h 30 FFFF 0464	Select 3Com name server socket
template define 9 h 12 FFFF 0600	Select XNS packet type
template define 10 h 12 FFFF 809B	Select Kinetics EtherTalk
template define 11 b 16 FF 15	Select AppleTalk source address 15
template define 12 b 17 FF 22	Select AppleTalk destination address 22
template define 13 h 12 FFFF 6004	Select DEC LAT packet type
template define 14 h 12 FFFF 6006	Select DEC DECnet packet type

In the following example, template 98 is defined to compare IP packets. This template causes packets to be compared in “half-word” (2-byte) chunks. The comparison to the template starts on the 13th byte (offset=12) of the packet. The mask and comparator are set, in a hexadecimal value, to compare certain portions of the packet.

```
59:ASN9000:bridge# template define 98 size=h off=12 mask=ffff comp=0000
Template 98: 12, 0xffff0000, 0x00000000: added
60:ASN9000:bridge#
```

**NOTE**

A full eight characters are expected when specifying the mask and comparator. If less than eight characters are specified, the end of the mask and comparators is padded with zeroes for the full eight characters.

3.2.2 Displaying Templates

The `config templates` command is used to display defined bridge templates. The following information is displayed:

- template number
- offset
- hexadecimal equivalent of the mask
- hexadecimal equivalent for the comparator

The following example displays the bridge template portion of the bridge configuration. The configuration displays three defined templates. The offset, mask and comparator values are shown for each template. (This example does not show the entire bridge configuration display).

```
69:ASN-9000:bridge# config templates
Filter templates
  Number  Offset  (dec)Mask (hex)  Comparator (hex)
  002     012    ff00ff00    00ff00f0
  098     024    0000ffff    0000ffff
  099     004    00000000    00000000
70:ASN-9000:bridge#
```

**NOTE**

Entries are displayed in numerical order to facilitate finding specific information. Keep in mind that the displayed order may not be the order in which the templates were defined or the order that they are applied against packets.

For more information about the `bridge config` command, refer to the *ForeRunner ASN-9000 Software Reference Manual*.

3.2.3 Deleting Templates

The `template undefine` command is used to delete a defined template. This command removes the template number from the defined templates. To modify a template it is necessary to delete the template and then to redefine it. The syntax for this command is:

```
template undefine <tnum>
```

where

<tnum> Specifies the template number. Valid template numbers range from 1 through 98. Only one template can be deleted at a time.



A bridge template cannot be deleted with the `template undefine` command until all rules applying the template have been deleted. Use the `lrule undefine` command to delete the rules applying the template.

The following example deletes template 98.

```
61:ASN9000:bridge# template undefine 98
Ok
62:ASN9000:bridge #
```

3.3 Rules

Rules are logical expressions containing templates and/or rules as operands. Up to 62 rules, 101 through 162, can be defined. An additional rule, rule 163, is a pre-defined rule which cannot be altered or deleted. Refer to *Section 3.3.1* for more information on rule 163. In addition to templates and rules, the rule logical expressions can contain any, or all, of the following logical expressions:

- Ampersand (&) to denote a logical AND operation.
- Vertical bar (|) to denote a logical OR operation.
- Tilde (~) to denote a logical NOT operation
- Parentheses (()) group operands in a complex expression.

Although the bridging software is designed to handle many nested rules, it is recommend that rules be made as brief as possible.

Rules are evaluated on packets and may terminate the evaluation of the packet early, before applying all templates and rules, as soon as it is determined that the packet should be filtered.

Evaluating a simple rule with one template adds about 5% to the total time required to process a packet. For example, if a typical packet requires two simple rules to be evaluated, the throughput in packets per second (pps) is about 90% of the rated maximum. If a typical packet requires four rules, each with two templates to be evaluated, then the throughput is approximately 0.95^8 or 66% of the rated maximum.

3.3.1 Pre-defined Rules

The bridge rule definition does not automatically send a packet that has evaluated to true through the entire series of filter rules. In order for such a packet to be sent through the network, the packet must match one last “match anything” rule. This rule (rule 163, containing template 99) is provided so that packets that have evaluated to true all the way through the filtering process can be sent to the rest of the network. (For information on template 99, refer to *Section 3.1*.) Rule 163 can be displayed using the `config rules` command. Rule 163 cannot be deleted or modified in any way.

3.4 Working with Rules

After individual templates are defined, rules that use those templates must be defined. Rules can be applied to source (incoming) or destination (outgoing) packets. When a rule evaluates to true, the packet being evaluated by the rule is filtered out.

Parentheses can be used to group template numbers, rule numbers and logical operators. Up to eight levels of parentheses can be used. The maximum total number of characters for a rule, including blanks, is 64.

3.4.1 Defining Rules

The `lrule define` command is used to define a rule which is be applied to a template. The syntax for this command is:

```
lrule define <rnum> <rule-statement>

      where
      <rnum>      Specifies the rule number. Valid rule number
                  numbers range from 101 - 162.
      <rule-statement> Specifies the template number assigned using the
                  template define command. The rule statement
                  can contain operators as well as template numbers.
```

Each rule consists of template numbers, or other rules, which can be joined by the logical operators `&` (AND), `|` (OR) or `~` (NOT). Parentheses can be used to group templates. Parentheses are optional unless used with multiple templates and multiple operators.

Table 3.2 provides some simple examples of rule definitions based on the template examples shown in Table 3.1. Notice that in the definition of `lrule 106` in Table 3.2, other rule s can be used in the equation defining a rule. With this in mind, it is possible to build some very complex rules on other, less complex, rules.

Table 3.2 - Rule Definition Examples

Command	Defines Rules to Filter Out...
<code>lrule define 101 (1&8&9)</code>	3Com name server broadcast
<code>lrule define 102 3 9</code>	TCP/IP or XNS packets
<code>lrule define 103 (3&(4 5))</code>	IP network address 55
<code>lrule define 104 10 & 11</code>	AppleTalk source address 15
<code>lrule define 105 10 & 12</code>	AppleTalk destination address 22
<code>lrule define 106 ~(104 105)</code>	AppleTalk source address 15 or destination address 22
<code>lrule define 107 1 & 13</code>	LAT broadcast packets
<code>lrule define 108 7 14</code>	DECnet or ARP packets

The following examples add rule 101 containing template 90, followed by rule 102 containing templates 90 and 98.

```
71:ASN9000:bridge# lrule define 101 90
Rule 101: 90 added
72:ASN9000:bridge# lrule define 102 90&98
Rule 102: 90&98 added
73:ASN9000:bridge#
```

In the second example, two templates are assigned to rule 102. The rules are joined with the AND operator (&) which forces packets to be compared to both template 90 and template 99 as one unit. The result of the comparison (`true` or `false`) depends on how the packet evaluates against both operands (the two template numbers). For example, if the packet matched template 90 or 98, but not both, the rule evaluates to `false`, and the packet is discarded. However, if the packet matches both templates, the rule evaluates to `true`, and the packet is sent to the rest of the network or to other filters (if defined). Notice that the template numbers and operators are not separated by spaces. Optionally, templates and operators can be enclosed in a single pair of parentheses, although parentheses are not required until multiple operators are combined, as shown in the following example:

```
73:ASN9000:bridge# lrule define 103 (90&98) | (90|98)
Rule 103: (90&98)|(90|98) added
74:ASN9000:bridge#
```

In this example, packets are compared to two conditions: template 90 and 98 as a whole, then template 90 or 98 individually. Because both conditions are separated by the OR symbol (`|`), if either of these comparisons is true, the packet proceeds to the rest of the network. In this rule, the packet is first compared against rules 90 and 98 as a whole. If the comparison evaluates to `true`, the packet is sent to the rest of the network. If the comparison evaluates to `false`, the packet is checked against template 90. If the packet evaluates to `false`, it is compared against template 98. If the comparison evaluates to `false` for all comparisons, then the packet is discarded. However if the packet evaluates to `true` for either the combination of template 90 and 98, or `true` for the comparison to either template 90 or 98, then the packet is permitted to proceed through the network.

3.4.2 Displaying Rules

After rules are defined, they can be viewed as part of the bridge configuration. The bridge configuration displays the configured bridge packets to nodes information. Part of the bridge configuration information is what templates, rules, and filters have been defined. The `config rules` command is used to display the defined rules. This command displays the following information:

- rule number
- description of the templates contained within each rule

Bridge Filters

The following example displays five defined rules and the templates that are used in each rule. Also, notice that the bridge configuration shows the “match anything” rule, rule 163.

```
74:ASN9000:bridge# config rules

Filter rules
  Number  Description
  101     90
  102     90&98
  103     (90&98)|(90|98)
  163     99
```

```
75:ASN9000:bridge#
```

For more information on the `config` command, refer to the *ForeRunner ASN-9000 Software Reference Manual*.

3.4.3 Deleting Rules

The `lrule undefine` command is used to delete a defined rule. To modify a rule, it is necessary to delete the rule with the `lrule undefine` command and then redefine the rule with the required changes. The syntax of this command is:

```
lrule undefine <rnum>
```

where

<rnum> Specifies the rule number being deleted. Only one rule can be deleted at a time.

The following example shows rule 103 being deleted.

```
76:ASN9000:bridge# lrule undefine 103
Rule 103 deleted
77:ASN9000:bridge#
```

3.5 Filters

Filters are the totality of the defined templates and rules. Templates and rules become a filter when applied to a segment. This step in creating a filter is what allows the templates and rules to begin functioning. Filtering does not occur until the templates and rules are actually attached to a segment.

Bridge packets are filtered according to the templates contained in the rule within the filter. How the software filters bridge packets can further be defined by specifying that packets be filtered according to the direction (transmit or receive) they are travelling. Directions (transmit or receive) are in relation to the ASN-9000. If the rule evaluation for a packet is `true`, the packet is dropped.



Only one receive rule and one transmit rule can be assigned to a segment. If a rule is assigned to a segment that already has a rule, the new rule overwrites the old one.

3.6 Working with Filters

Bridge rules can be attached to segments or to a specific MAC-layer address on a segment. Depending on the type of filter, the bridge table can be prevented from adding “learned” MAC-layer addresses.

3.6.1 Attaching Rules to Filters

The `filter attach` command is used to attach a filter to a segment. When attaching a rule to a filter, the direction (transmit or receive) on which the filter should process the bridge packets is specified as well as the segment. Although bridge filters are normally attached to a segment, bridge filters can also be attached to a MAC-layer addresses. These filters are called bridge node filters. To create a bridge node filter, use the `filter attach node` command. This command enables packets that are sent to or received from a specific MAC-layer address to be controlled.



Before a node filter can be created, the MAC-layer address to filter must be entered into the bridge table as a permanent bridge entry using the `bt add` command (refer to the *ForeRunner ASN-9000 Software Reference Manual ForeRunner* for more information on the `bt add` command).

Bridge Filters

Packets sent to or received from that node are recognized. When packets are destined for that MAC-layer address or queued to be transmitted from that address, the packets are passed through the filtering process before being sent. Therefore, packets do not occupy bandwidth on the segment until after the filtering process has determined whether the packets should be sent or discarded. Also, notice that the command object (**node**) is required, and that a blank space is required to introduce the Ethernet address that follows. The syntax for these commands is:

```
filter attach <rnum> receive|transmit <seglst>  
filter attach <rnum> node <ethaddr>
```

where

- <rnum>** Specifies the rule number assigned to this filter. Valid rule numbers are 101 through 162.
- receive|transmit** Specifies whether the rule is defined as receive or transmit. Receive rules cause bridge packets to be filtered as they are received on a segment. Transmit rules cause bridge packets to be filtered as they are transmitted on a segment.
- <seglst>** Specifies the segment to which the rule is applied. A single segment or a comma-separated list of segments can be specified.
- <ethaddr>** Specifies the MAC-layer address to which to attach the specified rule. The MAC address must be preceded with the **node** command object and can be entered as one contiguous 6-byte entry, each byte can be separated with a colon (:), or each byte can be separated with a dash (-).

In the following example, a receive rule is attached to segments 2.1 and 2.2. Therefore, all bridge packets received on segments 2.1 and 2.2 are sent through the filtering process before being accepted. Notice that a comma separates each segment, spaces are not required between the two segments.

```
78:ASN9000:bridge# filter attach 101 receive 2.1,2.2  
Receive rule 101 set for segment 2.1  
Receive rule 101 set for segment 2.2  
Rule 101 applied for segment 2.2 receive  
79:PowerHub:bridge#79:ASN9000:bridge#
```



In the preceding example, only the last rule applied was acknowledged. However, all the rules are applied. To list the rules that have been applied, issue the `config filters` command.

In the following example, a transmit rule is attached to segments 2.3 and 2.4. Therefore, all bridge packets are sent through the filtering process before being transmitted on segments 2.3 and 2.4.

```
79:ASN9000:bridge# filter attach 102 transmit 2.3,2.4
Transmit rule 102 set for segment 2.3
Transmit rule 102 set for segment 2.4
Rule 102 applied for segment 2.4 transmit
80:ASN9000:bridge#
```

The following example applies rule 101 on MAC address 00:00:ef:03:9a:b0.

```
95:ASN9000:bridge# filter attach 101 node 00:00:ef:03:9a:b0
warning: node rule will not take effect until a permanent bridge
table entry for 00:00:ef:03:9a:b0 is added.
96:ASN9000:bridge#
```

3.6.2 Displaying Filters

The `config filters` command is used to display all attached filters and respective rules. This command displays the following information:

- segment to which filters have been attached
- type of filter (transmit or receive)
- number of the transmit or receive rule

Bridge Filters

The following example shows three configured filters. One is a transmit while the other two are receive filters. The filters are listed in attached segment order.

```
90:ASN9000:bridge# config filters
```

```
Filters applied
  Segment      Transmit  Receive
    1.1         -         -
    .
    .
    .
    1.32        -         -
    2.1         101        -
    2.2         -         -
    2.3         -         102
    2.4         -         102
    2.5         -         -
    .
    .
    .
    3.1         -         -
```

```
91:ASN9000:bridge#
```



All segments are listed. Only those segments that have filters applied display either a transmit or receive rule.

3.6.3 Detaching Rules from Filters

The `filter detach` command is used to delete a filter. To modify a rule, it is necessary to first delete the rule and then to re-attach the rule with the desired changes. To detach a bridge node rule, issue the `filter detach node` command. This command removes the filters from the specified node address and, therefore, the node can send and receive packets without restriction. The syntax of these commands is:

```
filter detach      receive|transmit <seglist>
filter detach <rnum> node <ethaddr>
```

where

receive|transmit Specifies to detach a receive or a transmit rule from the specified segment or segment list.

<seglist> Specifies the segment, or segments, from which to detach the logical rule. One segment, or a comma-separated list of segments, can be specified.

- <rnum>** Specifies the rule number to be detached from the specified MAC address.
- <ethaddr>** Specifies the MAC-layer address from which to detach the rule specified by the *<rnum>* argument. The MAC address must be preceded with the **node** command object and can be entered as one contiguous 6-byte entry, each byte can be separated with a colon (:), or each byte can be separated with a dash (-).

The following example detaches the receive rule from segments 2.1 and 2.2.

```
33:ASN9000:bridge# filter detach receive 2.1
Receive rule 100 set for segment 2.1
Rule removed from segment 2.1 receive
34:ASN9000:bridge#
```

Bridge Filters

CHAPTER 4

Host Filters

The Host subsystem contains filter and template commands that are used to control the TCP and UDP packets sent to the ASN-9000. Host filters and templates do not address packets that are routed through to an eventual destination, only packets that are addressed to the ASN-9000 IP address are affected by host filters and templates.

Host filters are packet filters that allow or deny access based on the entire packet, as opposed to specific information contained within a packet. They are set up as the interfaces on which TCP and UDP packets are filtered. The commands presented in this chapter describe how to:

- Define/Undefined templates
- Display templates
- Define/Undefined filters
- Attach/Detach filters
- Display filters

4.1 Templates

Host templates are constructed by defining the desired conditions and assigning a number to those conditions. If multiple templates are set up in a filter, the software proceeds through the templates one-by-one in the order in which they are defined. Therefore, the template numbers provide a way of referencing the set of conditions, and do not imply the order in which the filters are applied to packets when filtering occurs.

4.2 Working with Templates

Templates must be constructed before filters. The template commands are used to:

- Define templates
- Display templates
- Delete templates

4.2.1 Defining Templates

The first step in creating a Host filter is defining a template. Multiple templates can be created for use by the same filter. However, each individual template must have a unique template number. Template numbers range from 1 to 32.

The `template define` command is used to define a Host template. The syntax of this command is:

```
template define <template-number> [sipa=<ipaddr>] [sipm=<ipaddr>]
[dipa=<ipaddr>] [dipm=<ipaddr>] [ipproto={tcp,udp}|<protonum>]
[tsport{=|<|>}<wks>|<portnum>] [tdport{=|<|>}<wks>|<portnum>]
[action=pass|block]
```



When specifying command objects, the equals sign (=) must be included as part of the syntax.

where

`template define <num>`

Specifies the template number. Valid numbers are from 1 to 32. The template number is a way of identifying the set of conditions. The template number does not imply the order in which templates are filtered.

`[sipa=<ipaddr>]`

Specifies the source IP address. The source device is the one sending the TCP or UDP packet. When the source IP address is specified, specify only the network portion. Specify the node portion of the address as all zeroes. For example, 147.128.0.0.

If the source IP address is specified, specify a mask (see `sipm`) for the source IP address. The source IP address is specified in either hex or dotted decimal notation.

`[sipm=<ipaddr>]`

Specifies the mask of the source IP address. The source mask must be valid for the IP address specified in the `sipa` object. When the source IP mask is specified, specify the amount of the address that should be used. For example, 255.255.0.0 specifies that the network portion of the address is

pertinent for filtering, whereas the node address portion is not. Specify the mask in hex or dotted decimal notation.

[dipa=<ipaddr>] Specifies the destination IP address. The destination device is the one receiving the TCP or UDP packet (the ASN-9000). When the destination IP address is specified, specify only the network portion. Specify the node portion of the address as all zeroes. For example, 147.128.0.0.

If the destination IP address is specified, specify a mask (see **dipm**) for the destination IP address. The destination IP address is specified in either hex or dotted decimal notation.

[dipm=<ipaddr>] Specifies the mask of the destination IP address receiving the TCP or UDP packet. The destination mask must be valid for the IP address specified in the **dipa** object. Specify the mask in dotted decimal notation (for example, 255.255.255.255), or in prefix notation (for example, /16).

[ipproto={tcp,udp}<pronum>] Specifies the protocol type for the packets being matched against the templates. Valid protocol types for host filters are **tcp** and **udp**. Optionally, a protocol number can be specified for the type of packets being matched. The protocol number is a well-known number as described in RFC 1340, “Well-Known Ports.” Specify the port numbers associated with TCP and UDP ports. For more information about the well-known ports, refer to the *ForeRunner ASN-9000 Software Reference Manual*.

[tspport{=<|>><wks><portnum>] Specifies the transport-layer protocol source port. Use this object only if the **ipproto** object is specified. This option only works for packets that match against the template when the **ipproto** object is set to **tcp** or **udp**.

Use the operators less than (<), greater than (>) or equal to (=) to enable a match on packets that are sent from certain “well-known” sockets or port numbers.

If `<wks>` is specified the well-known TCP socket on which the TCP or UDP packets are travelling must be specified. Alternatively, the well-known port name or number on which the TCP and UDP packets are travelling can be specified.

[tdport{=<|>><wks>|<portnum>}]

Specifies the transport-layer protocol destination port. Use this object only if the `ipproto` object is specified. This option works only on packets that match against the template when the `ipproto` object is set to `tcp` or `udp`.

Use the operators less than (`<`), greater than (`>`) or equal to (`=`) to enable a match on packets that are sent from certain “well-known” sockets or port numbers.

If `<wks>` is specified, the well-known TCP socket on which the TCP or UDP packets are travelling must be specified. Alternatively, the well-known port name or number on which the TCP and UDP are travelling can be specified.

[action=pass|block]

Specifies whether the template should pass or block the packet when the result of the template-to-packet comparison evaluates to true. If `pass` is specified, the packet is allowed access. If `block` is specified, the packet is prevented access. The default action is `block`.

In the following example, template 11 is defined based on a source IP address and mask. This template is constructed to match packets that are sent from source IP address 147.128.0.0. (The node portion of the IP address is not important in this command). This template is also set to mask out the last two bytes of the IP address so that the ASN-9000 knows that the network portion of the address is where the comparison to the packet must occur. Because no action was specified, the ASN-9000 uses the default action, `block`, and prevents packets that match the conditions set.

```
69:ASN9000:host# template define 11 sipa=147.128.0.0 sipm=255.255.0.0  
Ok. Template 11 defined.  
70:ASN9000:host#
```

The next example is very similar to the preceding one, except the destination IP address and mask are used as matching criteria for the packets being sent.

```
55:ASN9000:host# template define 32 dipa=147.128.0.0 dipm=255.255.0.0  
Ok. Template 32 defined.  
56:ASN9000:host#
```

In the next example, template 12 is defined to match TCP packets sent to the ASN-9000. Notice that the template has been constructed so that source and destination IP addresses are of no concern. The ASN-9000 matches only on the TCP packets sent to it, and because the action is **pass**, these packets are allowed to access.

```
56:ASN9000:host# template define 12 ipproto=tcp action=pass
Ok. Template 12 defined.
57:ASN9000:host#
```

4.2.2 Displaying Templates

The **template [show]** command is used to display the defined templates. The information related to the defined templates is displayed. The following information is displayed:

- number of the template
- source IP address and mask combination
- destination IP address and mask combination
- IP protocol type
- source port for TCP/UDP packets
- destination port for TCP/UDP packets
- whether the TCP packets sent are telnet connection request (**conreq**) packets
- action that must be taken when packets that match the conditions are encountered

The syntax of this command is:

```
template [show] [<template-number>]
```

where

[<template-number>] Specifies the template to display. Valid template numbers are from 1 through 32. If no template is specified, all defined templates are displayed.

The following example displays templates 12 and 32. Template 12 is set to pass any TCP packets that are sent from anywhere and template 32 is set to block any packets that are sent anywhere on IP address 147.138.0.0

```
74:ASN9000:host# template
T#          Source IP address/mask      Destination IP address/mask  ipproto
TCP/UDP source port      dest port    TCP conreq    Action
=====
12          any/any                any/any
                                     pass          TCP
-----
32          any/any                147.128.0.0/255.255.0.0
                                     block
```

```
75:ASN9000:host#
```



Templates are displayed in numerical order to facilitate finding a specific template number. This numerical order is only for display purposes; the templates are not necessarily defined in the order displayed.

4.2.3 Deleting Templates

The `template undefine` command is used to delete a template. When a template is deleted, the conditions set in the template are removed. To modify a template, issue this command then redefine the template with the appropriate changes. The syntax of this command is:

```
template undefine <template-number>
```

where

<template-number> Specifies the template number to delete. Only one template can be deleted at a time.

The following example deletes template 12.

```
76:ASN9000:host# template undefine 12
Ok. Template 12 undefined.
77:ASN9000:host#
```

4.3 Filters

After templates have been defined, they are associated with a filter number when the filter is defined. Filters are comprised of templates. Multiple templates can be contained within a single filter. When multiple templates are defined, the ASN-9000 processes the template in the order in which they have been defined. Filtering occurs:

- after templates have been defined
- when a filter number has been associated with the templates that constitute the filter
- when the filter is attached to a segment

4.4 Working with Filters

Once filters have been defined, the ASN-9000 can begin filtering. A maximum of 192 filters can be created.

During the filtering process, the packets proceed sequentially through the list of templates contained within each filter. When multiple filters are assigned, if a packet is not filtered by the first filter, the packet proceeds to the next filter where it is checked against the templates within that filter. This process continues until the packet is either discarded or processed through the last filter and accepted by the ASN-9000.

4.4.1 Defining Filters

The `filter define` command is used to create a filter. This command associates certain template(s) with a filter number.



To create a filter, at least one template must be specified with each filter number.

The syntax of this command is:

```
filter define <filter-number> <template-number>[,<template-number>...]
```

where

<filter-number> Specifies the number assigned to the filter. Valid filter numbers are from 1 through 192.

<template-number> Specifies the number of the template that is to become part within the filter.

[,<template-number>...] Specifies additional templates to become a part of the filter. If multiple templates are being defined, separate each with a comma.

In the following example, filter 192 is defined containing templates 12 and 32. Notice that the templates are separated with a comma.

```
3:ASN9000:host# filter define 192 12,32
4:ASN9000:host#
```

4.4.2 Attaching Filters

The `filter attach` command is used to attach a filter to a receive data stream segment or group of segments. This enables packets that are addressed to the ASN-9000 itself to be filtered. The syntax of this command is:

```
filter attach <rule-number> r[receive] <seglst>
```

where

- | | |
|----------------------------------|---|
| <code><rule-number></code> | Specifies the rule number of the filter to attach to the specified segment. Valid filter rule numbers are 1 through 32. |
| <code>r[receive]</code> | Specifies that the filter is to be applied to received packets. |
| <code><seglst></code> | Specifies the segment that should be checked. Specify one segment or a comma-separated list of segments. |

In the following example, filter 17 is defined. This is a receive filter attached to segments 2.10, 2.11 and 2.12. Packets received on this segments are checked.

```
6:ASN9000:host# filter attach 17 receive 2.10,2.11,2.12
7:ASN9000:host#
```

4.4.3 Displaying Filters

The `filter show` command is used to display defined filters. When this command is issued, the following information is displayed:

- number of each configured filter
- templates contained in the filter
- segments the filter is attached to

The syntax of this command is as follows:

```
filter [show] [f[ilter]=<filter-number>] [<seglist>]
```

where

f[ilter]= Specifies that a filter is to be displayed. If used, the equal sign (=) must be used.

[f[ilter]=<filter-number>] Specifies the filter number to be displayed. Specify one filter or a comma-separated list of filters. If multiple filters are specified, separate them with a comma. Valid filter numbers are 1 through 32.

If no particular filter is specified, all configured Host filters are displayed.

<seglist> Specifies the segments containing filters to be displayed. Specify one segment or multiple segments. If multiple segments are specified, separate them with a comma.

Host Filters

In the following example filter, 2 and 3, comprised of templates 11, 12 and 13 are defined. Also, Host receive filters 2 and 3 are defined on segments 1.4 and 1.5, respectively.

```
33:ASN9000:host# filter
```

```
Host Filter Template Definitions
```

Filter	Templates
1	11,12
2	12,13
3	11,13

```
Host Receive Filter attachments
```

Segment	Filter
1.4	2
1.5	3

```
34:ASN9000:host#
```

4.4.4 Detaching Filters

The **filter detach** command is used to detach a receive filter. To detach a filter, specify the segment to which it is currently attached. The syntax of this command is:

```
filter detach r[ceive] <seglst>
```

where

- r[ceive]** Specifies that a receive filter is to be detached.
- <seglst>** Specifies the segment from which the receive filter is to be detached. Specify one segment or a comma-separated list of segments.

In the following example, the receive filter is removed from segments 2.10 and 2.11. The “slot.segment” format is used, and multiple segments are separated with commas.

```
9:ASN9000:host# filter detach receive 2.10,2.11
Rule 17 unapplied from segment 2.10
Rule 17 unapplied from segment 2.11
10:ASN9000:host#
```

4.4.5 Deleting Filters

The `filter undefine` command is used to delete a defined filter. After templates have been associated with a filter number it is not necessary to delete the templates individually. When the filter is deleted, all associated templates are deleted. The syntax of this command is:

```
filter undefine <filter-number>
```

where

<filter-number> Specifies the number of the filter to be deleted. Only one filter can be deleted at a time.

In the following example, filter 12 is deleted, and all templates associated with filter 12 are removed from the template table.

```
21:ASN9000:host# filter undefine 12
22:ASN9000:host#
```

Host Filters

The *ForeRunner* ASN-9000 supports Internet Protocol (IP) filters to allow or disallow IP packets that are sent to or received from the ASN-9000. There are two types of IP filters:

- IP route filters
- IP packet filters

IP route filters control IP packets based on the routes contained within the packet. IP packet filters control IP packets addressed to or routed through the ASN-9000 by allowing or disallowing access based on the source or destination IP address, or well-known port number. The packets pass or block entire packets.

IP filters route IP packets sent through the ASN-9000 to another destination, or those that are addressed specifically to the ASN-9000.

IP filters are comprised of IP templates. IP templates are a series of user-defined conditions against which IP packets are compared. The templates match the packets against defined patterns.

IP filters are created by first defining a template, then assigning the template to a filter, and finally associating the filter with a segment. Each of these steps is explained in detail in the following sections. Filtering occurs when templates have been defined, associated with a filter number, and then the filter number has been associated with an IP network address. When a packet encounters a filter, the packet is checked against any and all templates that comprise that filter. IP filter commands allow the user to:

- Define templates
- Display templates
- Delete templates
- Define filters
- Attach filters
- Detach filters
- Display filters
- Delete filters

The IP filter commands are a part of the IP subsystem. For a complete listing of the IP subsystem commands, refer to the *ForeRunner ASN-9000 Protocols Reference Manual*.

5.1 Templates

IP templates are the building blocks of IP filters. Templates are a set of user-defined conditions that are matched against IP packets that are routed to or through the ASN-9000. The filtering software is constructed so that IP templates return a result when packets are compared to the templates. If the result of the comparison is true, then the packet is passed or blocked, whichever is specified when defining the template. If the result of the comparison is false, then the packet proceeds to the next template or filter, if defined.

Multiple templates can be defined for the same filter. In this case, templates are applied to a packet in the order in which they occur in the template definition. For example, if template 9, 1, and 12 are defined for a filter, the IP packet is applied to template 9, then template 1, and finally, template 12. As this example shows, the templates are compared to the packet individually.

The filtering software is constructed so that packets that do not match the templates are passed on to the next template or filter. If a packet does not match any of the templates, by default it is blocked. Therefore, if an IP packet is to be passed that doesn't match all of the templates, a template must be defined that passes on any unmatched packet, and assign this packet to the last filter in the filter list.

Filtering occurs after templates have been assigned a filter number, and that filter has been attached to a segment. When an IP packet is received, the packet is checked against any defined filters. If there is a filter defined, the packet is compared against each template in the filter. The template commands explained in this section enable the following functions to be performed:

- Define templates
- Display templates
- Delete templates



Because the default action is to block, if packets that survive the filtering process are to be forwarded, a “match anything” template with the default action of pass must be configured. This template, template 99, should be configured as the last template so that packets that survive the filtering process can proceed to their ultimate destination.

5.2 Working with Templates

When defining templates, a template number must be specified. One, or many, templates can comprise a filter. By default, templates block all packets that do not match the templates during the filtering process. Therefore, if all unmatched packets are to be passed, set up the last template as a match anything template. To do so, create a template with no matching criteria except the action of pass. Up to 256 IP templates are supported. To create a template some required information must be provided, but other information is optional. However, even the optional information has some rules. The syntax of the `template define` command provides further explanation about what information is required and what is optional.

5.2.1 Defining Templates

The `template define` command is used to define an IP filter template. This command enables a template number to be assigned a set of specified conditions. With this command, only the command itself and a template number are required. The other objects are optional. However, some objects have counterparts. If some objects are specified without their counterparts, an error message is presented. For example, if the `sipa` object is specified, the `sipm` object must also be specified. The syntax of this command is:

```
template define <template-number> [sipa=<ipaddr> sipm=<ipaddr>]
  [dipa=<ipaddr> dipm=<ipaddr>] [ipproto={tcp,udp}|<protonum>]
  [ipopt=srcrt] [tcptype=conreq] [tSPORT{=|<|>}<wks>|<portnum>]
  [tdport{=|<|>}<wks>|<portnum>] [action=pass|block]
```



When specifying command objects, the equal sign (=) must be included.

where

<template-num>

User-defined number from 1 through 128. If more than 128 templates are assigned to a filter, an error message is returned. Multiple templates can be assigned in any order. Template numbers provide an easy way to identify the templates in a filter; they do not imply an order of processing. The ASN-9000 warns if a template number is defined with no contents.

[sipa=<ipaddr>]	Specifies the source IP address of the network sending the packet. If this object is specified as part of the template, it must be paired with the sipm object.
[sipm=<ipaddr>]	Specifies the source IP mask. If this object is specified as part of the template, the sipa object must also be specified. This object supports two types of syntax: <ul style="list-style-type: none">• Dotted decimal notation. This method uses decimal integers, with periods separating each of the four bytes in the subnet mask. For example: 255.255.255.0• Prefix notation. This method uses a forward slash and a decimal number to specify the subnet mask's length (in bytes). The length is known as the <i>prefix length</i>. The template applies the prefix to the first bit of the first octet in the IP address to know where the subnet mask should be applied. An example of prefix notation is: /24
[dipa=<ipaddr>]	Specifies the destination IP address. The sipa object must also be specified.
[dipm=<ipaddr>]	Specifies the destination IP mask. If this object is specified as part of the template, the sipm object must be specified. This object supports the same types of syntax as the sipm object.
[ipproto={tcp,udp} <protonum>]	Specifies the transport-layer protocol packets to be filtered. TCP and UDP transport-layer protocol packets are supported: <p>If TCP or UDP is specified, then TCP or UDP packets are filtered.</p> <p>If protonum is specified, packets received from the "well-known" protocol number specified are filtered. For a list of the "well-known" protocol numbers, consult the <i>ForeRunner ASN-9000 Protocols Reference Manual</i>.</p>
[ipopt=srcrt]	Specifies the IP option. This object enables the source-route bit in the IP packet to be set. By using this object, the route table can be bypassed altogether, and the packet is forwarded to a specific next-hop router.

[tcptype=conreq] Specifies the type of TCP packet to block. When this object is specified, the first packet of a TCP connection request is accepted or rejected if the ASN-9000 is the target of the connection request.

The **conreq** option blocks the TCP SYN packet within the first TCP packet received. This causes the transmitting station to drop the remainder of the TCP based session.

[tsport{=<|>}<wks>|<portnum>] Specifies the transport-layer protocol source port. Use this object only if the **ipproto** object is specified. This option only works for packets that match against the template when the **ipproto** object is set to **tcp** or **udp**.

Use the operators less than (<), greater than (>) or equal to (=) to enable a match on packets that are sent from certain “well-known” sockets or port numbers.

If <wks> is specified the well-known TCP socket on which the TCP or UDP packets are travelling must be specified. Alternatively, the well-known port name or number on which the TCP and UDP packets are travelling can be specified.

[tdport{=<|>}<wks>|<portnum>] Specifies the transport-layer protocol destination port. Use this object only if the **ipproto** object is specified. This option works only on packets that match against the template when the **ipproto** object is set to **tcp** or **udp**.

Use the operators less than (<), greater than (>) or equal to (=) to enable a match on packets that are sent from certain “well-known” sockets or port numbers.

If <wks> is specified, the well-known TCP socket on which the TCP or UDP packets are travelling must be specified. Alternatively, the well-known port name or number on which the TCP and UDP are travelling can be specified.

[action=pass|block] Specifies the action that the template is to take if the template-to-packet comparison results in a value of true. If **block** is specified, then the packet is

discarded. If `pass` is specified, then the packet is accepted and processed for routing through to its destination. The default is `block`.

The following example defines template number 11 to match packets sent from source IP address 147.128.0.0. (The node portion of the IP address is not important in this command.) This template is set to mask out the last two bytes of the source IP address. The packet at the location of the network address is compared to the template. The template checks TCP packets that are telnet connection requests. Because no action is specified, the default action of `block` is applied to discard packets that match the conditions set.

```
69:ASN-9000:ip# template define 11 sipa=147.128.0.0 sipm=255.255.0.0 ipproto=tcp tcp-  
type=conreq  
Ok. Template 11 defined.  
70:ASN-9000:ip #
```

The following example defines template number 68 to match packets received from an address on network address 147.130.0.0. The source mask specifies that only the first two bytes in the four-byte IP address are matched against the packet. The `ipopt` object specifies that the routing table is not consulted to locate the router with the least number of hops to a destination; the packet is just sent to the directly-connected next hop router. Because no action is specified when the template is created, the default action `block` is applied to the packets.

```
257:ASN-9000:ip# template define 68 sipa=147.130.0.0 sipm=255.255.0.0 ipopt=srcrt  
action=pass  
Ok. Template 68 defined.  
259:ASN-9000:ip #
```

5.2.2 Displaying Templates

The `template show` command is used to display a list of the currently-defined IP templates. This command displays all defined template definitions or a specific template number. The following information relative to defined IP templates is displayed:

- template number
- source IP address and mask combination
- destination IP address and mask combination
- IP protocol type
- source port for TCP/UDP packets
- destination port for TCP/UDP packets
- whether TCP packets sent are telnet connection request (`conreq`) packets
- action to be taken when packets that match the conditions are found

The syntax of this command is as follows:

```
template [show] [<template-number>]

      where

<template-num>  Specifies the template to display. If a template
                 number is specified, the template definition for that
                 template is shown. If issued without a specific
                 template number, all templates are displayed.
```

In the following example, template number 68 is displayed. This template is set to pass any IP packets sent to (or through) the ASN-9000 from anywhere on IP network 147.130.0.0.

```
70:ASN9000:ip# template 68
T#      Source IP address/mask      Destination IP address/mask  ipproto
      TCP/UDP source port      dest port  TCP conreq  Action
=====
68      147.130.0.0/255.255.0.0      any/any      srcrt
                                     pass
-----

71:ASN9000:ip#
```

IP filters can be set to control the IP packets sent from or to a specified IP address. Also, IP filters can control TCP and UDP packets sent to the ASN-9000.

5.2.3 Deleting Templates

The `ip template undefine` command is used to delete an IP template. This command removes a currently-defined template from the template definition list. The syntax for this command is:

```
template undefine <template-number>

      where

<template-number>  Specifies the template number. Only one template
                   can be deleted at a time.
```

The following example deletes template 68.

```
257:ASN-9000:ip# template undefine 68
Ok.
258:ASN-9000:ip #
```

5.3 Working with Filters

The IP filter commands enable the creation and use of IP filters. The commands in the following section enable the templates created earlier to create and assign filters. After the filters have been created, the filtering process can begin.

5.3.1 Defining Filters

The `filter define` command is used to define a filter. Templates for the filter must already exist in order for the filter to be defined. The syntax of this command is as follows:

```
filter define <filter-number> <template-number>[,<template-number>...]
```

where

<code><filter-number></code>	Specifies the filter number. The filter (and the templates that comprise it) are referenced by this number. Valid filter numbers are 1 through 64.
<code><template-number></code>	Specifies the template to be included in the filter. The template must already exist to be included.
<code>[,<template-number>...]</code>	Specifies additional templates to be included in the filter. If additional templates are specified, separate each with a comma.



The `filter define` command requires a filter number and at least one template to create a filter.

The following example defines filter 64 comprised of templates 11 and 68.

```
257:ASN-9000:ip # filter define 64 11,68
258:ASN-9000:ip #
```

5.3.2 Displaying Filters

The **filter show** command is used to display a filter. When this command is issued, the defined filters are displayed. The following information pertinent to the configured IP filters is displayed:

- filter numbers
- templates that constitute each filter
- segments to which transmit and receive filters are attached

The syntax of this command is:

```
filter [show] [f[filter]=<filter-number>] [<seglist>]
```

where

[f[filter]=<filter-number>] Optional object indicating the filter to display. Notice that the entire word or just the first letter can be specified. Also, if this object is included, the equal sign (=) must be included.

[<seglist>] Specifies the segments associated with the filter. Specify one segment or a comma-separated list of segments. If multiple segments are specified, separate them with commas.

In the following example two filters (11 and 60) are configured. Filter 11 contains three templates as does filter 60. Filter 11 is a transmit filter attached to segments 1.4 and 1.5. Filter 60 is a receive filter attached to segment 1.2 and 1.3.

```
233:ASN-9000:ip# filter show
IP Filter Definitions
  Filter      Templates
  -----
  11          19,23,67
  60          101,122,198

IP Transmit Filter attachments
  Segment     Filter
  -----
  1.4         11
  1.5         11

IP Receive Filter attachments
  Segment     Filter
  -----
  1.2         60
  1.3         60
234:ASN-9000:ip#
```

5.3.3 Deleting Filters

The **filter undefine** command is used to delete a filter. This command removes an existing filter. When removing the filter, specify only the filter number. It is not necessary to delete the individual templates. The syntax of this command is:

```
filter undefine <filter-number>
```

where

<filter-num> Specifies the filter number. When the filter deleted, all templates that constitute the specified filter are deleted as well.

In the following example filter 64 is deleted.

```
257:ASN-9000:ip# filter undefine 64
258:ASN-9000:ip #
```

5.3.4 Attaching Filters to a Segment

IP filters are enabled to the datastream that is sent or received on a segment. Use the **filter attach** command to attach a filter. When a filter is attached to a segment, specify whether the filter is to be attached to the transmit or receive data stream and which segment, or segments, on which to configure the filter. The syntax of this command is as follows:

```
filter attach <filter number> r[eceive]|t[ransmit] <seglist>
```

where

<filter number> Specifies the filter number. Valid filter numbers are 1 through 64.

r[eceive]|t[ransmit] Specifies whether the filter is being applied to the receive or transmit data streams. The receive data stream is the one that is incoming relative to the ASN-9000. The transmit data stream is the one that is outgoing relative to the ASN-9000. One of each type of filter can be applied to a segment. Either spell out the command object or specify just the first letter of each word.

<seglist> Specifies the segment on which the receive or transmit filters are attached. Specify one segment or a comma-separated list of segments.

In the following example, filter 23 is attached to segment 1.2 as a receive filter, so IP packets that are received on segment 1.2 are matched against the templates in filter 23.

```
234:ASN-9000:ip# filter attach 23 receive 1.2
235:ASN-9000:ip#
```

5.3.5 Detaching Filters from a Segment

The `filter detach` command is used to detach a filter. Filters can be detached from the datastream being received or transmitted on a segment. When issuing this command, the kind of filter being detached must be specified as well as the segment on which the filter is configured. The syntax of this command is:

```
filter detach r[receive]|t[ransmit] <seglist>
```

where

- | | |
|------------------------------------|--|
| <code>r[receive] t[ransmit]</code> | Specifies whether to detach the filter from the receive or transmit data stream of a segment. When specifying this object, the entire word, or the first letter, can be entered. |
| <code><seglist></code> | Specifies the segment from which to detach the filter. One segment or a comma-separated list of segments can be specified. |

The following example detaches filter 23. All templates contained within filter 23 are deleted when the filter is detached. After filter 23 is removed, the IP packets received on segment 1.2 unrestricted.

```
23:ASN-9000:ip# filter detach 23 receive 1.2
24:ASN-9000:ip#
```

IP Filters

Internet Protocol/Routing Information Protocol (IP/RIP) Import and Export filters enable control of route information that is added to the route table. When routes are learned, either dynamically or statically, they are posted and entered into the route table for the route. When a packet is required to reach its destination, the route table is scanned to find the lowest cost route to that destination.

IP/RIP import and export filters enable a route's access to the route table to be permitted or denied (passed or blocked). Common usages of IP/RIP import and export filters are to control route information in the following situations:

- When routes are propagated to the rest of the network in RIP updates
- When routes are exported to the link-states configured on OSPF routers
- When routes are imported to the IP route table from OSPF link-state database advertisements.

By setting up export filters, route information that is being deleted from the route table can be controlled. For example, when RIP updates (packets which intermittently advertise the routes to destinations) are sent, an export filter can be set up so that only certain routes are advertised while others are hidden. Or, when connected to an OSPF router, an export filter would prevent some routes in the route table from being reported to the OSPF router's link-state database (for more information on OSPF, refer to the *ForeRunner ASN-9000 Protocols Reference Manual*). Export filters can be set to pass or block packets containing the information specified.

By setting up import filters, route information that is added to the route table can be controlled. For example, when a new router comes on line, a RIP update advertising the route to that router is received. By setting up import filters, the new router's information can be added to the route table or blocked from being added to the route table. Import filters can be set to pass or block packets containing specified information. In order to create IP/RIP import and export filters, first create a template, then assign the templates to a filter, which is applied to an interface. For more information about the IP/RIP protocol and the commands in the IP/RIP subsystem, see the *ForeRunner ASN-9000 Protocols Reference Manual*.

6.1 Templates

Templates are a set of user-defined conditions that are compared to a packet. When defining a filter first define templates to check the packet at a more granular, byte-by-byte level. When defining a filter the action that must be performed is specified when the templates match the packets, either pass or block. When the template is compared to the packet, the result of the comparison is either `true` or `false`. If the template evaluates to `true`, then the action specified is performed by the filter. When the result of the comparison is `false`, the packet continues on to other filters (if defined).

Up to 98 templates can be defined. Valid template numbers range from 101-199. Filters containing multiple templates can be created. If multiple templates are created, they are applied in the order in which they appear in the filter. For example, if the filter contained templates 101, 103, and 102, the packet is applied to template 101, then 103, and then 102. The number assigned to a template is a way to reference the particular set of conditions for that template. The template numbers do not imply any order of execution when filtering occurs. The template commands in the IP/RIP subsystem are used to:

- Define templates
- Display templates
- Delete templates
- Display template statistics
- Clear template statistics

6.2 Working with Templates

To create an IP template, assign a template number to the series of specified conditions. These conditions are what the packet is compared to during the filtering process. Filtering does not occur until the templates have been inserted into a filter and the filter has been associated with an interface or segment.

6.2.1 Defining Templates

The `template define` command is used to create a template. This command enables the conditions against which the packet is compared when filtering is actually occurring to be specified. The syntax of this command is:

```
template define <template-number> [rif=<ipaddr>] [target=<ipaddr>/
    <mask>] [gw=<ipaddr>/<mask>] [tif=<ipaddr>]
    [sproto=static|direct|rip|ripd|ospf] [tag=<tag>][tag\! =<tag>]
    action=[pass|block][,tag:<tag>][,metric:<metric>][,pref:<pref>]
```

where

<template-number>	Specifies the template number. Valid numbers are 101 through 198.
[rif=<ipaddr>]	Specifies the IP address on which packets are received.
[target=<ipaddr>/<mask>]	Specifies the IP address/mask of the route to be filtered.
[gw=<ipaddr>/<mask>]	Specifies the gateway IP address from where the route is learned.
[tif=<ipaddr>]	Specifies the IP address on which the packets are transmitted.
[sproto=static direct rip ripd ospf]	Specifies the source protocol that is to be filtered on the specified routes. If <code>static</code> is specified, the template matches packets that are sent over a statically configured IP route. If <code>direct</code> is specified, the template matches packets sent over interface (direct) routes. If <code>rip</code> is specified, the template matches packets sent over the RIP network. If <code>ripd</code> is specified, the template matches packets sent over default RIP routes. If <code>ospf</code> is specified, the template matches packets sent from an OSPF network.
[tag=<tag>][tag\! =<tag>]	Specifies a series of bytes that are appended to the packet. Tags are specified in hexadecimal format, and are eight bytes in length. For example, a valid tag is <code>0xffff0000</code> . Because tags are appended to the packet, they follow the packets through the network. Templates can be constructed to filter out packets that have specific tags appended.

Alternatively, the **tag!** argument can be specified. The **tag!** argument specifies that the tag does not equal a certain value.

```
action=[pass|block]
      [,tag:<tag>]
      [,metric:<metric>]
      [,pref:<pref>]
```

Specifies the action to be taken when the specified condition is met. If **pass** is specified the matched packets are allowed to pass through. If **block** is specified the packets are discarded. If **[,tag:<tag>]** is specified, the specified <tag> is appended to the packets. If **[,metric:<metric>]** is specified, the <metric> is used to modify outgoing metric on export filters. It must be a constant between 1 and 15. If **[,pref:<pref>]** is specified, the preference with which the route is added to the routing table (on import filters) is appended.



IP addresses and masks are entered as 192.168.0.0/255.255.0.0, 10.1.2.3/h (host route), or 10.0.0.0/8 (10.0.0.0/255.0.0.0). <tag> is a 32-bit hex number, optionally starting with '0x'. For gateway (gw) and target, the host portion of <ipaddr> is zeroed out.

In the following example, template 102 is defined to match RIP packets received on 147.128.136.70 carrying route information for network 148.150.0.0. The mask specifies to compare only the first two bytes of the four-byte network address. If these packets come by way of gateway 147.138.0.0, the packet should be blocked (the default action), because no action was specified.

```
17:ASN9000:ip/rip# template define 102 rif=147.128.136.70 target=148.150.0.0/25
5.255.0.0 gw=147.138.0.0/16
Ok. RIP filter template 102 defined.
18:ASN9000:ip/rip#
```

To export the static default route 0.0.0.0./0.0.0.0 into rip, use the following template:

```
9:PowerHub:ip/rip# template define 1 target=default sproto=static action=pass
Ok. RIP filter template 1 defined
```

6.2.2 Displaying Templates

The **template show** command is used to display configured templates. This command displays the template number and respective conditions. The syntax of this command is:

```
template [show] [<template-number>]
```

where

<template-number> Specifies the template number. Valid template numbers are 101 through 198.

In the following example, a template has been constructed to match RIP packets received on 147.128.136.70 carrying route information for network 148.150.0.0. The mask specifies to compare only the first two bytes of the four-byte network address. If these packets come by way of gateway 147.138.0.0, the packet should be blocked (the default action), because no action was specified.

```
46:ASN9000:ip/rip# template
Template #102:
  target=148.150.0.0/255.255.0.0
  gw=147.138.0.0/255.255.0.0
  rif=147.128.136.70
  action=default
```

```
RIP Template Count: 1
47:ASN9000:ip/rip#
```

6.2.3 Deleting Templates

The **template undefine** command is used to delete a template. This command enables template conditions to be selectively removed by deleting just the template number. One or all templates can be removed. The syntax of this command is:

```
template undefine <template-number>|all
```

where

<template-number> Specifies the template number. Specifying **all** causes all defined templates to be deleted.

The following example deletes template 1.

```
44:ASN9000:ip/rip# template undefine 1
Ok. RIP filter template 1 undefined.
45:ASN9000:ip/rip#
```

6.2.4 Displaying Template Statistics

The `template stats` command is used to display the current statistics for a template. How many times the template has been successfully matched, since the last clear, both for import and for export, is displayed. The syntax of this command is:

```
template [show] stats [<template-number>]
```

where

<template-number> Specifies the template number. Valid template numbers are 1 through 192.

The following example displays the number of packets that matched template 102. The columns of numbers indicate how many packets have been matched to the templates as they were imported to or exported from the route table.

```
50:ASN9000:ip/rip# template stats
                        Import  Export
Template #102:         0        0
51:ASN9000:ip/rip#
```

6.2.5 Clearing Template Statistics

The `template clear stats` command is used to clear the statistics for all templates in both the import and export filters. When clearing template statistics, they are reset to zero and begin incrementing again immediately after the reset. The syntax of this command is:

```
template clear stats
```

The following example clears the IP/RIP Import/Export filter stats. In reality, packets are transmitted and received so rapidly that these statistics do not remain at zero. Once the statistics have been reset to zero, they begin incrementing again, almost immediately.

```
53:ASN9000:ip/rip# template clear stats
Ok. Template Statistics cleared.
54:ASN9000:ip/rip#
```

6.3 Filters

Filters are the totality of the assigned templates. A filter is the logical grouping of templates applied to packets. Filters can be created only after the template(s) that should be checked against packets have been defined.

After the templates have been defined, assign a filter number to those templates that are to comprise the filter. The filter number is an easy way of referencing the templates that comprise the filter. Filter numbers can range from 101-198. Filtering can occur only after the templates have been defined, a filter number has been assigned to the packets, and specified whether the filters are import or export filters. The IP/RIP filter commands are able to:

- Define filters
- Display filters
- Insert filters
- Append filters
- Undefine filters
- Delete filters

For more information about IP/RIP, refer to the *ForeRunner ASN-9000 Protocols Reference Manual*.

6.4 Working with Filters

To create IP/RIP Import or Export filters, define the individual templates that are to be compared against packets, then associate those templates with a filter number. As part of specifying the filter, indicate what the filter should do with the packets that match the template. In the case of IP/RIP filters, specify import or export to allow or disallow RIP update packets from entering or exiting the IP route table.

6.4.1 Defining Filters

The `filter define` command is used to define an IP/RIP Import/Export filter. When defining a filter, assign a valid filter number, note which templates the filter contains, and specify whether the filter affects RIP updates entering or exiting the route table.



At least one template must be specified as part of the filter being defined.

The syntax of this command is:

```
filter define import|export <template-number>[,<template-number>...]
```

where

import|export Specifies the type of filter. Import filters perform the filtering process on RIP update packets that are received. Export filters perform the filtering process on the RIP packets that are sent.

<template-number> Specifies the number of the template to be associated with this filter.

[,<template-number>...] Specifies additional templates to add to the filter. One or multiple templates can be added. If multiple templates are added, separate each with a comma.

The following example defines an import filter containing templates 101 and 102.

```
79:ASN9000:ip/rip# filter define import 101,102
Ok.
80:ASN9000:ip/rip#
```

6.4.2 Displaying Filters

The **filter show** command is used to display defined filters. This command displays the defined import and/or export filters, and the templates associated with them. The syntax of this command is:

```
filter [show] [-d] [import|export]
```

where

[-d] Specifies to display the details of the defined import and/or export filters.

[import|export] Specifies the type of filter that to display. If **import** is specified, then all import filters are displayed. If **export** is specified, then all export filters are displayed. If neither is specified, all IP/RIP filters are displayed.

In the following example, all import and export filters are displayed. The `-d` option has been specified which causes all template details to be displayed.

```
89:ASN9000:ip/rip# filter -d
RIP Export filter: none defined

RIP Import filter: templates 101,102,123

Template #101:
target=169.144.0.0/255.255.0.0
gw=169.144.0.0/255.255.0.0
rif=169.144.86.55
action=default

Template #102:
target=148.150.0.0/255.255.0.0
gw=147.138.0.0/255.255.0.0
rif=147.128.136.70
action=default

Template #123:
target=148.151.0.0/255.255.0.0
gw=148.150.0.0/255.255.0.0
rif=148.111.200.65
action=default

90:ASN9000:ip/rip#
```

6.4.3 Appending Filters

The `filter append` command is used to append import or export filter templates to the template list for each filter. The syntax of this command is:

```
filter append import|export <template-number>[,<template-number>...]
```

where

import export	Specifies whether appending a template to an import or export filter.
<template-number>	Specifies the template number being appended.
[,<template-number>...]	Specifies additional templates that are being appended. If multiple templates are appended, separate each with a comma.

In the following example, template 123 is appended to the import filter.

```
86:ASN9000:ip/rip# filter append import 123
Ok.
87:ASN9000:ip/rip#
```

6.4.4 Inserting Filters

IP/RIP Import and Export filters have a unique feature, in that one or more templates can be inserted between existing templates in a filter. This feature is beneficial because it allows the order in which templates are applied to packets to be re-configured. For example, suppose templates 101, 102, and 103 are configured in a filter, and template 104 is to be added, after template 101. This can be done without having to redefine the filter. To insert a template, use the **filter insert** command. The syntax of this command is:

```
filter insert import|export before|after <template-number>|all
        <template-number>[,<template-number>...]
```

where

import|export Specifies whether the template to be inserted is part of an import or export filter.

before|after Specifies the target position of the templates being inserted. If **before** is specified, the template(s) are inserted before another template. If **after** is specified, the template(s) are inserted after another template.

<template-number>|all Specifies the template that is the reference point for the **before** or **after** argument. Specifying **all**, causes all templates to be relocated to the beginning or end of the template list for the filter.

<template-number>[,<template-number>...] Specifies the template to insert. One or multiple templates can be specified. If multiple templates are specified, separate each with a comma.

In the following examples, the currently defined templates are first displayed, followed by the current filter assignments. The current filter shows IP/RIP Export filter templates to be 101,102. The object is to insert templates 100 and 123 between 101 and 102. Template 100 is first inserted before template 102. The result shows the IP/RIP Export filter templates to now be 101,100,102. Then template 123 is inserted after template 100. The result shows the IP/RIP Export filter to contain templates 101,100,123,102. Command lines 105 and 107 could have been combined into one command by inserting templates 100 and 123 after 101 (i.e., **filter insert import after 101 100,123**).

```
103:ASN9000:ip/rip# template
Template #100:
target=129.165.0.0/255.255.0.0
gw=129.165.0.0/255.255.0.0
rif=129.165.2.45
action=default

Template #101:
target=169.144.0.0/255.255.0.0
gw=169.144.0.0/255.255.0.0
rif=169.144.86.55
action=default

Template #102:
target=148.150.0.0/255.255.0.0
gw=147.138.0.0/255.255.0.0
rif=147.128.136.70
action=default

Template #123:
target=148.151.0.0/255.255.0.0
gw=148.150.0.0/255.255.0.0
rif=148.111.200.65
action=default

RIP Template Count: 4
104:ASN9000:ip/rip# filter
RIP Export filter: none defined
RIP Import filter: templates 101,102
105:ASN9000:ip/rip# filter insert import before 102 100
Ok.
106:ASN9000:ip/rip# filter
RIP Export filter: none defined
RIP Import filter: templates 101,100,102
107:ASN9000:ip/rip# filter insert import after 100 123
Ok.
108:ASN9000:ip/rip# filter
RIP Export filter: none defined
RIP Import filter: templates 101,100,123,102
109:ASN9000:ip/rip#
```

6.4.5 undefining Filters

The **filter undefine** command is used to remove defined filters. When removing a filter, it is necessary to specify whether the filter is an import or an export filter. Because only one import and one export filter can be specified when this command is issued, the configured filter is removed. It is not necessary to remove each template assigned to the filter. The syntax of this command is:

```
filter undefine import|export
```

where

import|export Specifies either an import or export filter.

The following example removes the import filter.

```
77:ASN9000:ip/rip# filter undefine import
Ok.
78:ASN9000:ip/rip#
```

6.4.6 Deleting Filters

The **filter delete** command is used to delete filters that have been created and assigned. This command enables templates to be selectively removed from a filter.

The syntax of this command is:

```
filter delete import|export <template-number>[,<template-number>...]
```

where

import|export Specifies the type of filter template being deleted.

<template-number> Specifies the template number to delete from either the import or export filter.

[,<template-number>...] Specifies additional templates to be deleted from the filter. If multiple templates are specified, separate each with a comma.

In the following example, template 123 is removed from the import filter.

```
94:ASN9000:ip/rip# filter delete import 123
Ok.
95:ASN9000:ip/rip#
```

The PowerHub supports Internet Protocol/Open Shortest Path First (IP/OSPF) routing is supported on the *ForeRunner* ASN-9000. IP/OSPF routers contain a link-state database that is analogous to a route table. The link-state database in each OSPF router provides each router with a view of all routes to all destinations within the entire OSPF network.

IP/OSPF routers can be configured on the edge of the IP/OSPF network as area border routers. These routers can interface with other networks most commonly using RIP. Area border routers import RIP routes to their link-state database, and export link-state updates from OSPF (analogous to RIP updates) to the routers in the RIP network. For more information about the implementation of the IP/OSPF protocol, refer to the *ForeRunner ASN-9000 Protocols Reference Manual*.

IP/OSPF Export filters control link-state information that is exported from the OSPF link-state database. Exporting consists of removing link-state database information from the link-state database. Export filters allow, or disallow, OSPF information from propagating to RIP networks.

IP/OSPF Export filters are route filters. They perform their filtering on the route information contained within the packets that are exported to the RIP network. These filters are applied to the ASN-9000s that are configured as area border routers.

IP/OSPF export filters are comprised of templates. The first step in assigning an OSPF filter is to define the templates that check the packets that are sent from the link-state database table to the RIP networks. The commands in this chapter are used to configure IP/OSPF templates and filters.

7.1 Templates

Templates are used to compare user-defined conditions against packets that are sent or received. The templates determine which criterion is used for filtering. For example, packets can be filtered based on routes in the packet, among other conditions.

Templates are assigned numbers for reference. The template numbers can be from 1 to 192. They do not imply any order of processing during filtering. So, for example, if template 101, 102, 104, and 103 are assigned to a filter, the templates are processed in that order. In this example, template 104 is processed before 103.

Multiple templates can be assigned to one filter, and the same template can be assigned to multiple filters. If more than one template is assigned to a filter, the templates are processed sequentially. IP/OSPF template commands enable the following tasks to be performed:

- Define templates
- Display templates
- Delete templates
- Display template statistics
- Clear template statistics

7.2 Working with Templates

IP/OSPF templates consist of conditions that are applied to filter packets. When a packet is matched to the defined template(s), the link-state database update packets are either forwarded or discarded. The packets are discarded or forwarded based on the action specified when the template was created. Templates used in IP/OSPF filtering can be used to keep IP/OSPF network information separate from the route tables of the routers in the IP/RIP network.

7.2.1 Defining Templates

The first step in assigning a filter is creating a template. The filter references the templates by number. The template can be used in as many filters as necessary without needing to recreate the template again with each new filter. To create a template, use the `template define` command. The syntax of this command is:

```
template define <template-number> [target=<ipaddr>/<mask>]
    [gw=<ipaddr>/<mask>] [sproto=static|direct|rip|ripd|ospf]
[action=[pass|block][,tag:<tag>][,metric:<metric>][,pref:<pref>]
    [,type:<type>]]
```

where

<template-number>	Specifies the template number. Valid template numbers are 1 through 192.
[target=<ipaddr>/<mask>]	Specifies the IP route to look for when performing filtering. Optionally, a mask can be specified. Masks determine what portion of the network address to ignore and what portion to filter on. Use dotted-decimal notation when entering the IP address. A subnet mask can also be specified to determine how many bytes of the IP address are significant for the filtering process.
[gw=<ipaddr>/<mask>]	Specifies the IP address or subnet mask of the gateway router. Use dotted-decimal notation when entering the IP address. A subnet mask can also be specified to determine how many bytes of the gw address are significant for the filtering process.
[sproto=static direct rip ripd ospf]	Specifies the protocol type of the packet to filter. This condition enables the selection of protocol-specific packets to filter on a specific interface instead of filtering all packets that traverse that interface. If static is specified, packets that contain routes that have been statically assigned are filtered. If direct is specified, packets that are sent over interface (direct) routes are filtered. If rip is specified, packets received from the RIP network are filtered. If ripd is specified, packets received from the default RIP routes are filtered. If ospf is specified, packets sent from the OSPF network are filtered.

<tag> is a 32-bit hex number, optionally starting with '0x'. On export route-filters, <metric> can be used to modify the outgoing metric. It must be a constant between 1 and 65535. The type can be used to set the external type to 1 or 2. On import filters <pref> can be used to assign the preference with which the route will be added to the routing table. NOTE: For gateway(gw) and target, host portion of <ipaddr> is zeroed out

<pre>[action=[pass block] [,tag:<tag>] [,metric:<metric>] [,pref:<pref>] [,type:<type>]]</pre>	<p>Specifies the action to be taken with the packets that match the template conditions. If pass is specified, the packets are allowed. If block is specified, the packets are discarded. Those packets that are allowed can have information appended to them which can be filtered for in other templates. These include: [,tag:<tag>], [,metric:<metric>], [,pref:<pref>] or [,type:<type>]. Tags are eight-byte numbers that are associated with a packet and follow the packet through the network. Tags provide another way for to match against template conditions. Tags are always specified in hexadecimal notation. As an option, the tag can begin with "0x." This notation indicates that the value is a hexadecimal value.</p>
--	--

In the following example, template 13 is defined to match RIP packets sent through the gateway router on network 147.129.0.0. The slash notation (/16) indicates that the first 16 bits of the 32-bit network address to the RIP packet that is being exported is compared.

```
129:ASN9000:ip/ospf# template define 13 gw=147.129.0.0/16 sproto=rip
Ok. OSPF filter template 13 defined.
130:ASN9000:ip/ospf#
```

To export the static default route 0.0.0.0/0.0.0.0 into OSPF, use the following template:

```
11:PowerHub:ip/ospf# template define 1 target=default sproto=static action=pass
Ok. OSPF filter template 1 defined.
```

7.2.2 Displaying Templates

Templates that have been defined can be displayed by using the **template show** command by specifying the number of the template to display. If no template is specified, all defined templates are displayed. The syntax of this command is:

```
template [show] [<template-number>]
```

where

<pre><template-number></pre>	<p>Specifies the template number. If no template number is specified, all templates are displayed.</p>
------------------------------------	--

In the following example, templates 12 and 13 are displayed.

```
146:ASN9000:ip/ospf# template
Template # 12:
target=148.130.0.0/255.255.0.0
gw=147.130.0.0/255.255.0.0
sprto=static
action=pass

Template # 13:
gw=147.129.0.0/255.255.0.0
sprto=rip
action=default
```

```
OSPF Template Count: 2
147:ASN9000:ip/ospf#
```

7.2.3 Deleting Templates

To delete templates, use the `template undefine` command. When this command is issued, it is only necessary to specify the template number, or `all`, to delete all templates. It is not necessary to remove the template conditions set with the `template define` command. The syntax of this command is:

```
template undefine <template-number>|all
                        where
<template-number>|all  Specifies the template number. If all is specified, all
                        defined templates are deleted.
```

In the following example, template 12 is deleted.

```
151:ASN9000:ip/ospf# template undefine 12
Ok. OSPF filter template 12 undefined.
152:ASN9000:ip/ospf#
```

7.2.4 Displaying Template Statistics

Statistics that are generated through filtering can be displayed by using the `template [show] stats` command. When this command is issued, the number packets matched against the template conditions is displayed. This information can be used to determine whether the template is operating properly. The syntax of this command is:

```
template [show] stats [<template-number>]
```

where

stats Is required to display template statistics. If the word **stats** is not specified, the OSPF templates configured are displayed.

[<template-number>] Specifies the template number for which to display statistics. If no template number is specified, then all statistics for all templates is displayed.

In the following example, statistics for template 13, the only template configured, are displayed.

```
153:ASN9000:ip/ospf# template stats
                Import  Export
Template # 13:      0      0
154:ASN9000:ip/ospf#
```

7.2.5 Clearing Template Statistics

Template statistics can be cleared by using the **template clear stats** command. When this command is issued, the template statistics are all set to zero. However, because packets are continuously being forwarded, the statistics begin incrementing. For this reason, template statistics are rarely at zero. The syntax of this command is:

```
template clear stats
```

In the following example, all IP/OSPF template statistics are cleared. Once the statistics are cleared, the filtering process begins to increment the template statistics immediately.

```
154:ASN9000:ip/ospf# template stats clear
Ok. Template Statistics cleared.
155:ASN9000:ip/ospf#
```

7.3 Filters

Once the templates have been created, filters can be created to determine what to do with the result of the packet-to-template comparison. Packets matching this comparison are filtered depending on the type of filters created. IP/OSPF filters either pass or discard packets containing OSPF routes exported to routers in the IP/RIP networks.

7.4 Working with Filters

IP/OSPF Import/Export filters are created by assigning a filter number to the previously defined templates. When an OSPF route is exported to, or imported from, an IP/RIP network, the link-state advertisement packet is checked. The following actions can be performed with these packets:

- The packet is allowed to be imported/exported if the link-state advertisement packet doesn't contain data that matches the template.
- The packet can be passed, or discarded, if the link-state advertisement packet does contain data that matches the template

7.4.1 Defining Filters

IP/OSPF Import/Export filters are defined using the **filter define** command. This command references an ordered list of templates which are used to check against the link-state database packet. The filtering process is applied against packets that are imported/exported to/from the OSPF link-state database table to other routers in the OSPF network, or other routers in an IP/RIP network. Any template matching the specified action is executed. The first action that includes 'pass' or 'block' terminates the search. The syntax of this command is:

```
filter define import|export <template-number>[,<template-number>...]
```

where

import export	Specifies whether this is an import or export filter.
<template-number>	Specifies the template number. Valid template numbers are 1 through 192.
,<template-number>	Specifies the additional template numbers to assign to the filter. If multiple templates are assigned to the same filter, separate each template with a comma.

In the following example, an export filter containing template 14 is defined.

```
172:ASN9000:ip/ospf# filter define export 14
Ok.
173:ASN9000:ip/ospf#
```

7.4.2 Appending Filters

OSPF Export filters offer the unique ability to append templates to pre-defined filters. You can append filters by using the `filter append export` command. This command enables you to add one or more templates to the end of the template list for an existing filter without having to delete the current filter and completely redefine it with the new templates. The syntax of this command is as follows:

```
filter append import|export <template-number>[,<template-number>...]
```

where

<code>import export</code>	Specifies whether to append the template to an import or export filter.
<code><template-number></code>	Specifies the number of the template to append. Valid template numbers are 1 through 192.
<code>,<template-number></code>	Specifies additional templates to append. If multiple templates are specified, separate them with commas.



As many templates as desired can be appended, as long as the total number of templates does not exceed 192.

In the following example, template 121 is appended to the end of the list of templates applied to the configured export filter.

```
198:ASN9000:ip/ospf# filter append export 121
Ok.
199:ASN9000:ip/ospf#
```

7.4.3 Inserting Filters

IP/OSPF filters can have templates inserted between other templates. To insert a template, use the `filter insert` command. This command enables templates to be rearranged to change the order in which they are processed. This enables templates to be repositioned without having to delete the current filter and redefine the filter with new templates.



Templates are processed sequentially, so the order in which the templates occur in the filter is the order in which they are applied to the packet.

The objects **before** and **after** are provided to specify the location (in relation to the templates already in the filter) to append one or more templates. Additionally, it is possible to combine the **before** and **after** objects with the keyword **all**. By doing so, one or more templates can be positioned at the front or end of the template list. For example, template 192 could be placed at the front of the template definition by issuing this command with the **before all** combination, and template 1 could be placed at the end of the filter by issuing an **after all** combination.



The **before** and **after** objects are mutually exclusive; it is not necessary to specify both “**before 10**” and “**after 9**.” Stating either places the template in the proper position.

The syntax of this command is:

```
filter insert import|export before|after <template-number>|all
        <template-number>[,<template-number>...]
```

where

import export	Specifies whether to insert the template in an import or export filter.
before after	A positional object. Specifying before positions the template before another template. Specifying after positions the template after another template.
<template-number> all	Specifies the template referenced as the target template. The template being inserted is positioned before or after this template. If all is specified, the specified template is inserted at the beginning or end of the template list.
<template-number>	Specifies the template being inserted. Valid template numbers are from 1 through 192.
[,<template-number>]	Specifies additional templates to be insert. If multiple templates are specified, separate them with commas.

In the following example, template 13 is inserted before template 121 in the list of templates for the configured export filter.

```
205:ASN9000:ip/ospf# filter insert export before 121 13
Ok.
206:ASN9000:ip/ospf#
```

7.4.4 Displaying Filters

The `filter [show] export` command displays the configured import, export or all defined filters. The syntax of this command is:

```
filter [show] [-d] [import|export]
```

where

[-d] Specifies to display the details of the displayed filters.

[import|export] Specifies to display either the import or export filters. If import or export is not specified, all defined filters are displayed.

In the following example, details of all defined filters and templates are displayed.

```
208:ASN9000:ip/ospf# filter -d
OSPF Export filter: templates 14,13,121
```

```
Template # 14:
  target=147.129.0.0/255.255.255.255
  gw=147.128.0.0
  sproto=static
  action=pass
```

```
Template # 13:
  gw=147.129.0.0/255.255.0.0
  sproto=rip
  action=default
```

```
Template #121:
  gw=147.129.0.0
  sproto=rip
  action=block
```

```
OSPF Import filter: none defined
```

```
209:ASN9000:ip/ospf#
```

7.4.5 undefining Filters

IP/OSPF filters can be removed by issuing the **filter undefine** command. This command enables an import or export filter to be removed. It is not necessary to remove each of the individual templates assigned to the filter. The syntax of this command is:

```
filter undefine import|export
```

In the following example, the export filter is removed from the defined filters list.

```
211:ASN9000:ip/ospf# filter undefine export
Ok.
212:ASN9000:ip/ospf#
```

7.4.6 Deleting Filters

Templates associated with an IP/OSPF import or export filter can be deleted by using the **filter delete** command. When this command is issued, the specified template is deleted from the import or export filter. The syntax of this command is:

```
filter delete import|export <template-number>[,<template-number>...]
```

where

import export	Specifies to delete the templates from the export filter.
<template-number>	Specifies the template being deleted. Valid template numbers are from 1 through 192.
[,<template-number>]	Specifies additional templates to delete. If multiple templates are specified, separate them with commas.

In the following example, template 121 is deleted from the export filter.

```
222:ASN9000:ip/ospf# filter delete export 121
Ok.
223:ASN9000:ip/ospf#
```


CHAPTER 8

IPX RIP/SAP Filters

Internetwork Packet Exchange Routing Information Protocol/Service Advertising Protocol (IPX RIP/SAP) filters provide security control over route and server information sent and received by IPX networks associated with ASN-9000 segments. Depending on the level of security desired, the following can be accomplished:

- Block a segments sending RIP or SAP updates for a particular network
- Block or allow route or server information to be sent or received on a specific network
- Block servers from being reported in response to an IPX “Get Nearest Server” request

IPX SAP filters restrict connectivity to IPX servers by controlling the receipt and transmission of SAP updates. Using SAP filters, secured servers can be “hidden” from workstations that attempt to connect to them. All filters apply to specific networks or servers on a specific segment.

IPX RIP filters restrict connectivity to IPX networks by selectively controlling the routes that are reported or accepted in RIP updates.

There are three different types of IPX RIP and SAP filters. This chapter describes how each type of filter works and the commands used to create, display, or delete them.

8.1 RIP and SAP Filter Types

Unlike the other protocol filters, the IPX RIP and SAP filters do not require templates. Filters are defined and assigned to segments or interfaces on the ASN-9000. The three different types of IPX RIP and SAP filters are:

- | | |
|---------------------------|--|
| Data Input Filter | Operates on the receiving end of the RIP or SAP update. When an IPX network on a specific segment receives a RIP or SAP update, data input filters accept or reject information in the update. |
| Data Output Filter | Operates on the sending end of the RIP or SAP report. Before the report is sent for an IPX network on a specific segment, data output filters allow specific entries to be reported in or discarded from the report. |

Packet Output Filter Operates on the sending end of the RIP or SAP update. Before the update for an IPX network is sent on a specific segment, packet output filters allow the entire packet to be reported or discarded.

8.1.1 Exclusivity

Filters of the same type (data-input, data-output or packet-output) are mutually exclusive. If a filter is defined that explicitly receives or sends specific information, all other information is implicitly discarded. For example, if a RIP data-input filter is defined that explicitly accepts RIP updates from a specific IPX network, all other RIP updates are blocked. To accept additional RIP updates, additional filters need to be defined.

If secure access is needed to just a few networks or servers, it's generally easier to define filters that block or discard update information sent by those networks or servers. However, if the network requires tight security, filters can be defined that explicitly allow only specific updates to be sent or received.

8.1.2 Entering IPX RIP and IPX SAP Commands

The RIP and SAP filter commands use the same syntax, but are entered in different subsystems. RIP commands are entered in the `ipx/rip` subsystem while SAP commands are entered in the `ipx/sap` commands. The command prompt indicates the current subsystem. Always make sure of the correct subsystem before entering commands. Otherwise, the desired filters can not be applied. To display all subsystems, use the `subsystems | ss` command. To change to a subsystem, enter the name of the desired subsystem.

8.1.3 Types of Control

IPX RIP and SAP filters provide different types of control. Depending upon the types of filters defined, they can filter according to the following:

- Network and interface combination
 - Data input and data-output filters, selectively filter according to specific networks on specific interfaces.
 - Data-input filters, selectively accept or discard updates sent from specific interfaces on a specific network.
 - Data-output filters, selectively send or block updates from a specific interface on a specific network.
- Segment and interface combination
 - Packet-output filters, selectively filter according to specific interfaces on specific segments. Updates to a specific interface on a specific segment can selectively be filtered to block or accept data.

8.2 IPX RIP Filters

IPX RIP filters control route information that is propagated on IPX networks. By selectively allowing RIP packets to circulate through the network, or trapping these packets and discarding them, network security can be provided. IPX RIP filters particular networks where the RIP packets should undergo the filtering process to be specified. The networks that are to be the target of IPX RIP filters, and the segment on which the ASN-9000 should watch for those packets that contain the specified network number can be specified. For example, block all RIP packets sent or received on segment 1.10, except those packets that contain route information for network 11223344. In this example, all packets on segment 1.10 are discarded, but those that contain route information for network 11223344.

Three different kinds of IPX RIP filters are supported:

- Data Input filters
- Data Output filters
- Packet Output filters

8.2.1 Data Input Filters

IPX RIP updates are received from other routers in the IPX network. Data input filters can be configured to pass or block IPX RIP update packets. Data input filters can be assigned to particular segments to restrict the filtering to those update packets received from specific areas of the network. Data input filters can:

- Add filters
- Display filters
- Delete filters

8.2.1.1 Adding Filters

Use the `data-input-filter add` command to create a data input filter. The conditions that the incoming packets must match can be specified, as well as the action to be taken when packets that match the specified filters are encountered. The syntax of this command is:

```
data-input-filter|dif add <fnum> b[lock]|p[ass] <targetnet> <rxnet>
```

where

<fnum>	Specifies the filter number. Valid filter numbers are from 1 through 128.
b[lock] p[ass]	Specifies the action the filter is to perform when it finds packets that match. If block is specified, then the route entry in the packet that matches the conditions in the filter is discarded. If pass is specified, the route entry in the packet that matches the filter is allowed to proceed through the network or to other filters (if defined).
<targetnet>	Specifies the IPX network to filter. If a network is specified, when a RIP packet that contains the specified network is found, the action (pass or block) specified is performed.

IPX network numbers are entered in decimal notation. The network number can be up to eight characters (alphabetic or numeric) in length. IPX network numbers are expected to be eight characters long. If a network containing less than eight characters is specified, the remaining characters are padded with zeroes.

Also, 0x can be specified before the IPX network number to indicate that the address is being entered in hexadecimal format. By default, decimal entries are expected. When 0x is specified for hexadecimal values, the IPX network address is displayed in decimal format so the address is easier to recognize.

<rxnet> Specifies the network on which RIP packets for the specified route information should be analyzed. This argument specifies the receive network, so the filter checks only the RIP updates received on the network specified.

The following example shows the syntax of this command:

```
3:ASN9000:ipx/rip# data-input-filter add 127 pass 55aacc55 33ccdd33
Ok
4:ASN9000:ipx/rip#
```

8.2.1.2 Displaying Filters

Use the **data-input-filter show** command to display the filter conditions for configured data input filters. The following information is displayed:

- filter number
- action the filter is to take when encountering packets that match
- route network number (the network number of the route that is to be filtered)
- network number on which the route network number is received

The syntax of this command is:

```
data-input-filter | dif [show] [<fnum-list>|all]
```

<fnum-list>|all Specifies the filters to display. Specify a single filter, or a comma-separated list of filters. If **all** is specified, then all configured data-input filters are displayed. Valid filter numbers are 1 through 128.

The following example shows the syntax of this command:

```
6:ASN9000:ipx/rip# data-input-filter
RIP Data Input Filters:
Fil Action Route-NW Rcvd-NW
---
120 block 11abcd22 33abcd44
127 pass 55aacc55 33ccdd33
7:ASN9000:ipx/rip#
```

8.2.1.3 Deleting Filters

Use the `data-input-filter delete` command to delete a data input filter. A data input filter can only be modified by deleting the existing filter and defining a new filter. The syntax of this command is:

```
data-input-filter|dif delete <fnum-list>|all
```

where

`<fnum-list>|all` Specifies the filter(s) to delete. Specify a single filter or a comma-separated list of filters. If `all` is specified, then all defined data-input filters are deleted. Valid filter numbers are from 1 through 128.

The following example shows data input filter 20 being deleted from the PowerHub IPX RIP filter definition.

```
10:ASN9000:ipx/rip# data-input-filter delete 120
Ok
11:ASN9000:ipx/rip#
```

8.2.2 Data Output Filters

IPX RIP reports are sent to other routers in the IPX network. Data output filters can be configured to pass or block IPX RIP report packets being sent to certain networks. Data output filters can be assigned to particular IPX networks so that filtering of report packets that are transmitted to specific areas of the network are restricted. Data input filters can:

- Add filters
- Display filters
- Delete filters

8.2.2.1 Adding Filters

When data output filter are defined, conditions that the outgoing packets must match are specified. Data output filters are created by issuing the `data-output-filter add` command. What the filter must do with the packets that match the IPX RIP filter is specified when creating the filter. IPX RIP filters do not require templates. You create all the necessary information in one step when creating the filter. The syntax of this command is:

```
data-output-filter|dof add <fnum> b[lock]|p[ass] <targetnet> <txnet>
```

where

<fnum>	Specifies the number assigned to the filter. Valid filter numbers are from 1 through 128.
b[lock] p[ass]	Specifies the action the filter is to perform when it encounters packets that match. If block is specified, then the route entry contained in the packet that matches the conditions in the filter is discarded. If pass is specified, then the route entry contained in the packet and the packet proceeds to the next hop in the network, or on to other filters (if they have been defined) are allowed to pass.
<targetnet>	Specifies the IPX network to filter. If a network is specified, when a RIP packet that contains the specified network is found, the filter performs the action (pass or block) specified.

When entering an IPX network number, enter the network number in decimal notation. The network number can be up to eight characters (alphabetic or numeric) in length. Because an IPX network number eight characters long is expected, if a network less than eight characters long is specified, the remaining characters are padded with zeroes to fill the network number out to eight characters.

Also, `0x` can optionally be specified before the IPX network number to indicate that the address is in hexadecimal format. By default decimal values are expected. When specifying `0x` for the hexadecimal value of the IPX network address, the address is displayed as decimal numbers.

<txnet> Specifies the IPX network that is transmitting the RIP packets. Only transmitted packets on the IPX network specified are analyzed.

The following example shows data output filter 122 is being added. This filter is a pass filter; allowing the transmission of IPX RIP packets containing route updates for network 44dd33cc on IPX network number 22bb11aa.

```
13:ASN9000:ipx/rip# data-output-filter add 122 p 44dd33cc 22bb11aa
Ok
14:ASN9000:ipx/rip#
```

8.2.2.2 Displaying Filters

Use the **data-output-filter show** command to display a list of the defined data output filters. This command shows the following:

- filter number
- action that the switch takes when it encounters packets that match the filter
- route network number (the network number of the route that is to be filtered)
- network number on which the route network number is transmitted

The syntax of this command is:

```
data-output-filter | dof [show] [<fnum-list>|all]
```

where

<fnum-list>|all Specifies the filters to display. Specify a single filter, or a comma-separated list of filters. If **all** is specified, then all configured data-output filters are displayed. If no filters are specified, all filters are displayed. Valid filter numbers are 1 through 128.

The following example, two data output filters are configured: filter 120 and filter 122. Filter 120 is a block filter preventing RIP update packets that contain route 11aa22bb from being sent on network 33cc44dd. The second filter, filter 127, is a pass filter allowing RIP updates that contain route 44dd33cc to be sent on network 22bb11aa.

```
17:ASN9000:ipx/rip# data-output-filter
RIP Data Output Filters:
Fil Action Route-NW Report-NW
--- -----
120 block 11aa22bb 33cc44dd
122 pass 44dd33cc 22bb11aa
18:ASN9000:ipx/rip#
```

8.2.2.3 Deleting Filters

Use the `data-output-filter delete` command to delete data output filters. To modify a data output filter, it is necessary to delete the existing filter and redefine a new filter. The syntax of this command is:

```
data-output-filter | dof delete <fnum-list> | all
```

where

<fnum-list>|all Specifies the filter(s) to delete. Specify a single filter or a comma-separated list of filters. If `all` is specified, then all data-output filters are deleted. Valid filter numbers are from 1 through 128.

The following example deletes the two data output filters shown above. The two filters are separated by only a comma.

```
18:ASN9000:ipx/rip# data-output-filter delete 120,122
Ok
19:ASN9000:ipx/rip#
```

8.2.3 Packet Output Filters

Packet output filters provide a way to control RIP updates that are sent to an entire network on a specific segment. These filters can be applied so that all RIP updates are sent or not sent to network on a segment. They are very similar to data-output filters, except that they block or pass updates to an entire network, instead of parts of a network.

8.2.3.1 Adding Filters

Use the `pkt-output-filter add` command to create a packet output filter. This command enables a specified segment to look for packets, the conditions against which packets are matched, the action to take when packets that match the filter are encountered, and assign the filter to a segment. The syntax of this command is:

```
pkt-output-filter | pof add <fnum> b[lock] | p[ass] <segment> <txnet>
```

where

<fnum> Specifies the filter number assigned to the filter. Valid filter numbers are from 1 through 128.

b[lock]|p[ass] Specifies the action to perform when packets that match the filter are encountered. If `block` is specified, all packets that match the conditions in the

filter are discarded. If **pass** is specified, the packets are allowed to proceed to the next hop in the network, or on to other filters (if defined).

<segment> Specifies the segment to monitor for the specified packets. Enter the segment in the segment numbering format (**<slot.segment>**). For example, segment 2 of slot 1, would be noted as follows:
1.2

<txnet> Specifies the IPX network that is transmitting the packets. Only transmitted packets on the IPX network specified are analyzed.

In the following example, packet output filter 15 is a pass filter allowing IPX RIP report packets to transmit to network 44ccdd44 on segment 1.5.

```
6:ASN9000:ipx/rip# pkt-output-filter add 15 p 1.5 44ccdd44
Ok
7:ASN9000:ipx/rip#
```

8.2.3.2 Displaying Filters

Use the **pkt-output-filter show** command to display the configured packet output filters. The following information is displayed:

- filter number
- action to be taken when encountering packets that match
- segment on which the IPX RIP report packets are sent.
- network that is contained within the IPX RIP report

The syntax of this command is:

```
pkt-output-filter|pof [show] [<fnum-list>|all]
```

where

<fnum-list>|all Specifies the filters to display. Specify a single filter, or a comma-separated list of filters. If no filter number is specified, all filters are displayed. If **all** is specified, then all filters are displayed.

In the following example, two filters configured: filters 14 and 15. Filter 14 is a block filter preventing IPX RIP report packets from being transmitted on segment 1.6 to network 55ccdd55. Filter 15 is a pass filter allowing IPX RIP report packets to be transmitted on segment 1.5 to network 44bbdd44. Since no filter was specified, all filters are displayed.

```
10:ASN9000:ipx/rip# pkt-output-filter
RIP Packet Output Filters:
Fil Action Segment Report-NW
---
14 block 1.6 55ccdd55
15 pass 1.5 44bbdd44
11:ASN9000:ipx/rip#
```

8.2.3.3 Deleting Filters

Use the `pkt-output-filter delete` command to delete or modify a filter. This command removes a defined packet output filter. To modify a filter, the filter must first be deleted and then redefined. The syntax of this command is:

```
pkt-output-filter|pof delete <fnum-list>|all
```

where

`<fnum-list>|all` Specifies the filters to delete. Specify a single filter or a comma-separated list of filters. If `all` is specified, then all packet-output filters are deleted.

In this example, packet output filter 15 is deleted.

```
7:ASN9000:ipx/rip# pkt-output-filter delete 15
Ok
8:ASN9000:ipx/rip#
```

8.3 IPX SAP Filters

IPX SAP is used to inform routers on the IPX network of the servers that are available. SAP updates received on the ASN-9000 are basically lists of the servers that are known to the router that originates the update. SAP reports can be sent to inform other routers of all the servers that the ASN-9000 knows about. By using IPX SAP filters, the type of server information that is propagated through the IPX network can be selectively controlled. Three different kinds of IPX SAP filters are supported:

- Data Input filters
- Data Output filters
- Packet Output filters

8.3.1 Data Input Filters

IPX SAP data-input filters block or pass SAP updates received by the ASN-9000. These filters prevent SAP updates from recording the server information to the IPX server table. Filtering is accomplished by matching the server type and server name in IPX SAP reports received on a specified interface. For example, these filters inform the ASN-9000 “of all SAP reports received on segment 1.10, pass or block those reports containing server type ‘X’ and server name ‘Y.’” IPX SAP data input filters allow:

- Adding filters
- Deleting filters
- Displaying filters

8.3.1.1 Adding Filters

Use the `data-input-filter add` command to create an IPX/SAP data input filter. When a data input filter is created, the conditions that the incoming packets must match and what the filter is to do with the packets that match the IPX SAP filter is also specified. Data output filters do not require the use of templates. The syntax of this command is:

```
data-input-filter|dif add <fnum> block|pass <stype> <sname> <rxnet>
```

where

<fnum>	Specifies the number assigned to the filter. Valid filter numbers are 1 through 128.
b[lock] p[ass]	Specifies the action to be performed when packets that match the filter are encountered. If block is specified, then all packets that match the conditions of the filter are discarded. If pass is specified, then packets that match the filter are allowed to report the servers in the server table, or on to other filters (if defined).

<stype> Specifies the server type to filter. Table 8.1 lists the valid server types:

Table 8.1 - Server Types

Mnemonic	Hex Equivalent
PRINT-QUEUE	0003
FILE-SERVER	0004
JOB-SERVER	0005
PRINT-SERVR	0007
ARCHIVE-SERVR	0009
REM-BRIDGE	0024
ADVRT-PRINT	0047

Enter the mnemonic value or the hexadecimal equivalent. If the number for the server type is used, enter it as shown above.

<sname> Specifies the name of the server. An individual server can be specified or an asterisk (*) can be entered to apply the filter to all servers of the specified type.

<rxnet> Specifies the network on which SAP packets are analyzed for server information. This argument causes only packets received on this network to be analyzed.

In the following example, data-input filter 20 is configured on the PowerHub. This filter is a block filter. It prevents the PowerHub from receiving SAP updates for the file server “engineering” on network 55ccdd55.

```
6:ASN9000:ipx/sap# data-input-filter add 20 block 0004 engineering 55ccdd55
Ok
7:ASN9000:ipx/sap#
```

8.3.1.2 Displaying Filters

Use the `data-input-filter show` command to display the data-input-filters currently defined. This command displays the following information:

- filter number
- action taken when packets matching the filter are encountered
- network on which the IPX SAP report packets are received
- server type contained within the IPX SAP packet
- server name contained within the IPX SAP packet

The syntax of this command is:

```
data-input-filter|dif show [-f] [<fnum-list>|all]
```

where

[-f] Specifies to display the full name of the server (default is first 39 chars).

[<fnum-list>|all] Specifies the filters to display. A single filter, or a comma-separated list of filters can be specified. If no filter number is specified, all filters are shown. If **all** is specified, then all data-input filters configured are displayed. Valid filter numbers are 1 through 128.

In the following example, data input filter 20 is configured. This filter is a block filter preventing IPX SAP packets from being received on network 55ccdd55 containing server information for the file server “engineering.”

```
8:ASN9000:ipx/sap# data-input-filter
SAP Data Input Filters:
Fil Action  Rcvd-NW  Server-Type  Server-Name
-----
 20 block   55ccdd55  FILE-SERVER  engineering
9:ASN9000:ipx/sap#
```

8.3.1.3 Deleting Filters

Use the **data-input-filter delete** command to delete a data-input filter. To modify a filter, it is necessary to delete the filter and then redefine it. The syntax of this command is:

```
data-input-filter|dif delete <fnum-list>|all
```

where

<fnum-list>|all Specifies the filter(s) that to delete. A single filter or a comma-separated list of filters can be specified. If **all** is specified, all data-input filters defined are deleted. Valid filter numbers are from 1 through 128.

In the following example data-input filter 23 being deleted:

```
10:ASN9000:ipx/sap# data-input-filter delete 20
Ok
11:ASN9000:ipx/sap#
```

8.3.2 Data Output Filters

IPX SAP reports can be sent to other routers in the IPX network. The reports contain a list of the servers known to the ASN-9000. Data output filters can be configured to pass or block the IPX SAP report packets received from certain networks. Data output filters can be assigned to particular IPX interfaces so that the filtering of packets are restricted to specific areas of the network. With data output filters the following can be accomplished:

- Add a filter
- Show a filter
- Delete a filter

8.3.2.1 Adding Filters

Use the **data-output-filter add** command add a data output filter. This command displays the following information:

- filter number
- action taken when packets that match the filter are encountered
- network on which the IPX SAP report packets are sent
- server type contained within the IPX SAP packet
- server name contained within the IPX SAP packet

The syntax of this command is:

```
data-output-filter | dof add <fnum> block|b|block-nearest|bn|pass|p  
                    <stype> <sname> <txnet>
```

where

<fnum>	Specifies the filter number assigned to the filter. Valid filter numbers are from 1 through 128.
block b block-nearest bn pass p	Specifies the action the filter is to perform when packets that match are encountered. If block is specified, all the packets that match the conditions are discarded. If block-nearest is specified, the server specified is hidden from being reported in response to IPX Get Nearest Server requests. Thus, the server is hidden from workstations on the specified networks, but is still reported to other routers in the network. If pass is specified, packets are allowed to proceed to the next hop in the network, or on to other filters (if defined).
<stype>	Specifies the server type to filter. Refer to Table 8.1 for a list of valid server types: The mnemonic value or the hexadecimal equivalent can be entered. If the number is used for the server type, enter it as shown in Table 8.1.
<sname>	Specifies the name of the server. An individual server can be specified or enter an asterisk (*) to apply the filter to all servers of the specified type.
<txnet>	Specifies the network on which SAP packets are analyzed for the specified route information. This command object causes only packets transmitted on this network to be analyzed.

In the following example, filter 32 is added as a pass filter allowing IPX SAP reports that contain information about the server “marketing” to be transmitted on network 11232.

```
13:ASN9000:ipx/sap# data-output-filter add 32 pass 0005 marketing 11232  
Ok  
14:ASN9000:ipx/sap#
```

8.3.2.2 Displaying Filters

Use the `data-output-filter show` command to display the configured data-output-filters. The syntax of this command is as follows:

```
data-output-filter|dof show [-f] [<fnum-list>|all]
```

where

[-f]

[<fnum-list>|all] Specifies the filters to display. A single filter, or a comma-separated list of filters can be specified. If **all** is specified, all the data-output filters are displayed. Valid filter numbers are 1 through 128.

In the following example, filter 29 is configured as a pass filter for the job server “marketing.” This filter allows IPX SAP packets with server information about “marketing” to be sent on network 11232.

```
14:ASN9000:ipx/sap# data-output-filter
SAP Data Output Filters:
Fil Action  Report-NW Server-Type Server-Name
-----
32 pass     11232      JOB-SERVER marketing
15:ASN9000:ipx/sap#
```

8.3.2.3 Deleting Filters

Use the `data-output-filter delete` command to delete data output filters. To modify a data output filter, it is necessary to first delete the filter and then to redefine it. The syntax of this command is:

```
data-output-filter|dof delete <fnum-list>|all
```

where

<fnum-list>|all Specifies the filter(s) to delete. A single filter or a comma-separated list of filters can be specified. If **all** is specified, then all the filters are deleted. Valid filter numbers are from 1 through 128.

In the following example, data-output filter 32 is deleted.

```
15:ASN9000:ipx/sap# data-output-filter delete 32
Ok
16:ASN9000:ipx/sap#
```

8.3.3 Packet Output Filters

Packet-output filters can be created to prevent entire packets from exiting the PowerHub ASN-9000. Packet output filters pass or block IPX SAP packets on specified segments that are destined for a specific network.

8.3.3.1 Adding Filters

Use the `pkt-output-filter add` command to create packet output filters. Templates are not required for IPX SAP filters. The syntax of this command is:

```
pkt-output-filter|pof add <fnum> block|pass <segment> <txnet>
```

where

- | | |
|------------------------|---|
| <fnum> | Specifies the filter number. Valid filter numbers are 1 through 128. |
| b[lock] p[ass] | Specifies the action the filter is to perform when packets that match the condition are encountered. If block is specified, all packets that match the conditions are discarded. If pass is specified, then the packets are allowed to proceed to the next hop in the network, or on to other filters (if defined). |
| <segment> | Specifies the segment on which to listen for specified packets. Enter the segment in the <code><slot.seg></code> format. For example, segment 2 residing in slot one, would be noted as:
1.2 |
| <txnet> | Specifies the IPX network transmitting the packets. Only transmitted packets on the IPX network specified are analyzed. |

In the following example, filter 38 is added to the PowerHub. This filter is a block filter. It prevents the PowerHub from transmitting packets on segment 1.5 to network 33aabb33.

```
42:ASN9000:ipx/sap# pkt-output-filter add 38 block 1.5 33aabb33
Ok
43:ASN9000:ipx/sap#
```

8.3.3.2 Displaying Filters

Use the `pkt-output-filter show` command to display the defined packet output filters. This command displays the following information:

- filter number
- action the filter must take when it matches against a packet
- segment on which the filter must analyze packets
- network to which packets are transmitted

The syntax of this command is:

```
pkt-output-filter|pof show [<fnum-list>|all]
```

where

[<fnum-list>|all] Specifies the filters to display. A single filter, or a comma-separated list of filters can be specified. If **all** is specified, all the packet output filters are displayed. Valid filter numbers are 1 through 128.

In the following example, filters 38 and 39 are configured. Filter 38 is assigned to block packets containing information on network 33aabb33 exiting on segment 1.5. Filter 39 is assigned to pass packets containing information on network 33aadd33 on segment 1.6.

```
45:ASN9000:ipx/sap# pkt-output-filter
SAP Packet Output Filters:
Fil Action Segment Report-NW
---
 38 block    1.5    33aabb33
 39 pass     1.6    33aadd33
46:ASN9000:ipx/sap#
```

8.3.3.3 Deleting Filters

Use the `pkt-output-filter delete` command to delete packet output filters. To modify a filter, it is necessary to delete a packet output filter and then redefine it. The syntax of this command is:

```
pkt-output-filter|pof delete <fnum-list>|all
```

where

<fnum-list>|all Specifies the filter(s) to delete. A single filter or a comma-separated list of filters can be specified. If **all** is specified, then all filters are deleted. Valid filter numbers are from 1 through 128.

IPX RIP/SAP Filters

In the following example, filter 38 is deleted.

```
48:ASN9000:ipx/sap# pkt-output-filter delete 38  
Ok  
49:ASN9000:ipx/sap#
```

Index

A

ampersand (&)	3 - 9
AND operation	3 - 9
AppleTalk filters	1 - 1, 2 - 1
adding forward filters	2 - 3
adding zone data input filters	2 - 9
adding zone data output filters	2 - 11
adding zone packet output filters	2 - 6
deleting forward filters	2 - 6
deleting zone data input filters	2 - 10
deleting zone data output filters	2 - 13
deleting zone packet output filters	2 - 8
exclusivity	2 - 2
input and output filters	2 - 2
introduction	2 - 1
multiple filters	2 - 2
NBP forward filters	2 - 3
network range	2 - 1
showing forward filters	2 - 4
showing zone data input filters	2 - 10
showing zone data output filters	2 - 12
showing zone packet output filters	2 - 7
zone data input filters	2 - 8
zone data output filters	2 - 11
zone packet output filters	2 - 6
zones	2 - 1

B

bridge filters	1 - 1, 1 - 2, 3 - 1
templates	1 - 3, 1 - 4
bridge node filters	3 - 13
bridge rules	3 - 9
163	3 - 9

Bridge Templates	3 - 2
template 99	3 - 4

F

filter	
output	1 - 1, 1 - 2
filters	
AppleTalk	1 - 2
data input filters	1 - 1
data output filters	1 - 1
IPX RIP	1 - 2
IPX SAP	1 - 2
packet output filters	1 - 2
types of	1 - 1

H

host filters	1 - 1, 4 - 1
attaching a filter	4 - 8
defining a filter	4 - 7
defining a template	4 - 2
deleting a filter	4 - 11
deleting a template	4 - 6
detaching a filter	4 - 10
showing a filter	4 - 9
showing a template	4 - 5
templates	4 - 1

I

input filter	1 - 1
IP filters	1 - 1, 5 - 1
attaching a filter to a segment	5 - 10
defining a filter	5 - 8
defining a template	5 - 3
deleting a filter	5 - 10

Index

- deleting a template 5 - 7
- detaching a filter from a segment . . . 5 - 11
 - packet 5 - 1
 - route. 5 - 1
- showing a filter 5 - 9
- showing a template. 5 - 6
- templates. 5 - 2
- IP/OSPF filters 1 - 1
- IP/RIP export and import filters 6 - 1
 - appending a filter 6 - 9
 - clearing template statistics. 6 - 6
 - defining a filter. 6 - 7
 - defining a template 6 - 3
 - deleting a filter 6 - 12
 - deleting a template 6 - 5
 - inserting a filter 6 - 10
 - removing a filter 6 - 12
 - showing a filter 6 - 8
 - showing a template. 6 - 4
 - showing template statistics 6 - 6
 - templates. 6 - 2
- IP/RIP filters 1 - 1
- IPX RIP and SAP filters 1 - 1
 - control 8 - 3
 - data input filters. 8 - 1
 - data output filters 8 - 1
 - entering commands. 8 - 2
 - exclusivity. 8 - 2
 - packet output filters 8 - 2
 - types of 8 - 1
- IPX RIP filters 8 - 3
 - adding data input filters. 8 - 4
 - adding data-output filters 8 - 7
 - adding packet output filters. 8 - 9
 - data input filters. 8 - 4
 - data output filters 8 - 6
 - deleting data input filters. 8 - 6
 - deleting data-output filters 8 - 9
 - deleting packet output filters. 8 - 11
 - packet output filters 8 - 9
 - showing data input filters 8 - 5
 - showing data-output filters 8 - 8
 - showing packet output filters 8 - 10
- IPX SAP filters. 8 - 11
 - adding data input filters. 8 - 12
 - adding data output filters 8 - 15
 - adding packet output filters. 8 - 18
 - data input filters. 8 - 12
 - data output filters 8 - 15
 - deleting data-input filters. 8 - 15
 - deleting data-output filters 8 - 17
 - deleting packet output filters. 8 - 19
 - packet output filters 8 - 18
 - showing data-input filters 8 - 14
 - showing data-output filters 8 - 17
 - showing packet output filters 8 - 19
- L**
 - logical filtering
 - checks 3 - 1
 - effect on performance 3 - 9
 - rules. 3 - 1
- N**
 - NOT operation 3 - 9
- O**
 - operators
 - logical
 - AND 3 - 9
 - NOT 3 - 9
 - OR. 3 - 9
 - parentheses 3 - 9
 - OR operation 3 - 9
 - OSPF filters 7 - 1
 - appending a filter 7 - 8

clearing template statistics	7 - 6
defining a filter	7 - 7
defining a template	7 - 3
deleting a filter	7 - 11
deleting a template	7 - 5
inserting a filter	7 - 8
removing a filter	7 - 11
showing a filter	7 - 10
showing a template	7 - 4
showing template statistics	7 - 5
templates	7 - 1
P	
performance	
effect of logical filtering on	3 - 9
PX RIP and SAP filters	8 - 1
R	
rules	
effect on performance	3 - 9
nested	3 - 9
S	
security filters	1 - 1, 1 - 3
bridge filters	1 - 2
route filters	1 - 2
rules	1 - 4
T	
template	
logical filtering	3 - 1
templates	1 - 3
tilde (~)	3 - 9
V	
vertical bar ()	3 - 9

Index