Release Notes for IronWare Software Release 03.0.01c

FastIron Edge Switch X-Series

FastIron Workgroup Switch X-Series

FastIron SuperX Switch

FastIron SX 800 Switch

FastIron SX 1600 Switch

Release date: September 21, 2006

Publication date: September 21, 2006

FOUNDRY NETWORKS

4980 Great America Parkway
Santa Clara, CA 95054
Tel 408.207.1700
www.foundrynetworks.com

These release notes contain the features, enhancements, and known software issues in software release 03.0.01c for the Foundry FastIron X-Series devices.

These release notes describe the following:

- “Addenda to Foundry FastIron X-Series Configuration Guide” on page 2
- “Feature Support” on page 6
- “Software Fixes” on page 82

These release notes also describe software releases 03.0.01b, 03.0.01a, 03.0.01, and 03.0.00:

- “Summary of Enhancement in 03.0.01a” on page 2
- “Summary of Enhancements in 03.0.01” on page 2
- “Summary of Enhancements in 3.0.00” on page 3
- “Feature Support” on page 6
- “Software Fixes” on page 82

NOTE: These release notes contain the terms **FastIron Edge Switch** (**FES**), **FastIron Edge Switch X-Series** (**FESX**), **FastIron SuperX**, **FastIron SX**, and **FastIron Workgroup Switch X-Series (FWSX**). Each term refers to a specific set of devices, as shown in Table 1.

Table 1: FastIron Product Family

This Name	Refers to These Devices
FastIron Edge Switch (FES)	FES2402, FES4802, FES9604, FES12GCF, FES2402-POE, and FES4802-POE
FastIron Edge Switch X-Series (FESX or FES X-Series)	FESX424, FESX424HF, FESX424-POE, FESX448
FastIron SuperX	FSX
FastIron SX	FSX 800 and FSX 1600
FastIron Workgroup Switch X-Series (FWSX)	FWSX424 and FWSX448

The FESX, FSX, FSX 800, FSX 1600, and FWSX are collectively referred to throughout these release notes as the FastIron X-Series devices.

NOTE: These release notes contain information specific to the above devices, and describe features that differ from the FES and the BigIron Chassis devices.

About IronWare Release 03.0.01c

This release is designed for the following Foundry devices:

- FastIron Edge Switch X424 (FESX424)
- FastIron Edge Switch X424 HF (FESX424HF)
- FastIron Edge Switch X424-POE (FESX424-POE)
- FastIron Edge Switch X448 (FESX448)
- FastIron Workgroup Switch X-Series 424 (FWSX424)

- FastIron Workgroup Switch X-Series 448 (FWSX448)
- FastIron SuperX Switch (FSX)
- FastIron SX 800 Switch (FSX 800)
- FastIron SX 1600 Switch (FSX 1600)

NOTE: You cannot use this software on the FastIron Edge Switch, other Foundry Stackable devices, or other Foundry Chassis devices.

NOTE: These release notes do not describe how to upgrade a base model (Foundry device running Layer 2 and base Layer 3 code) to a premium model (Foundry device running full Layer 3 code). To perform this upgrade, you need an upgrade kit. Contact Foundry Networks for information.

NOTE: The FWSX is a Layer 2 switch only. It does not support base Layer 3 nor full Layer 3 features.

Addenda to Foundry FastIron X-Series Configuration Guide

Feature	Description	See Page
Filtering OSPF routes based on the network mask of the destination network	Use an extended ACL with an OSPF distribution list to filter OSPF routes based on the network mask of the destination network.	20
Load balancing differences	This section describes how the FastIron X Series load balances unknown unicast, multicast, and broadcast traffic.	21

Summary of Enhancement in 03.0.01a

Enhancement	Description	See Page
Pressing the Enter after a Message of the Day is displayed is no longer required.	If a Message of the Day is configured, users no longer need to press the Enter key before the login prompt is displayed.	22

Summary of Enhancements in 03.0.01

System-Level Enhancements in 03.0.01

Enhancement	Description	See Page
New start date for daylight savings time	The software will automatically change the system clock to daylight savings time on the second Sunday of March in 2007. This enhancement complies with the new federally mandated start of daylight savings time, beginning in 2007.	N/A

Enhancement	Description	See Page
Specifying the maximum number of entries supported in the RMON control table.	Starting in software release 03.0.01, the maximum number of entries supported in the RMON control table, such as alarms, history, and events, has increased. In addition, you can manually specify the maximum number of entries allowed in the RMON control table.	23

Summary of Enhancements in 3.0.00

New Hardware in 03.0.00

Release 03.0.00 introduces the following hardware:

- FSX 800 chassis
- FSX 1600 chassis
- 2500 watt power supply for the FastIron SuperX chassis devices (not supported on stackable devices)

These items are described in the *Foundry FastIron X-Series Chassis Hardware Installation Guide*.

Layer 3 Enhancements in 03.0.00

Enhancement	Description	See Page
Outbound rate shaping	This feature can be used to shape the rate and to control the bandwidth of outbound traffic on a port.	31
VSRP, VRRP, and VRRP-E scale timer	The **scale-timer** command allows you to adjust the VRRP and VRRP-E timers for Hello interval, Dead interval, Backup Hello interval, and Hold-Down Interval.	32
VRRP-E slow start timer	The VRRP-E slow start timer causes a specified amount of time to elapse between the time a VRRP-E Master router comes back up and when it takes over from a Backup router. This interval allows time for OSPF convergence when the Master is restored.	33
Clearing OSPF information from the Foundry device	You can clear specific kinds of information from the Foundry device’s OSPF link state database and OSPF routing table. You do not need to remove statements from the Foundry device’s configuration or reload the software for the commands to take effect.	33
BGP null0 routing	BGP can use the null0 route to resolve its next hop. Thus, a null0 route in the routing table (for example, static route) is considered as a valid route by BGP.	39
Policy-based routing	Policy-Based Routing (PBR) allows you to use ACLs and route maps to selectively modify and route IP packets in hardware.	44
802.3ad support in the base Layer 3 code	The Foundry device now supports 802.3ad in the base Layer 3 code.	N/A

Layer 2 Enhancements in 03.0.00

Enhancement	Description	See Page
Additional MRP ring IDs	You can configure an MRP ring ID to be a number between 1 and 1023.	23
MRP Phase 2	In Metro Ring Protocol (MRP) Phase 2, the same physical interface can be shared by multiple rings belonging to the same VLAN.	23
Changes to spanning tree port priority	The spanning-tree port priority configuration is changed in accordance with the IEEE 802.1w standards.	30
VSRP-aware security	VSRP-aware security parameters protect VSRP-aware switches against unauthorized VSRP hello packets.	52

Security Enhancements

Restricting remote access to the device by using hardware filtering	ACLs can be used to create management filters. Adding management ACL entries into existing ACLs used for traffic filters can also be used to deliver this feature. All ACLs are applied in hardware. Refer to the *Foundry FastIron X-Series Configuration Guide* for instructions on how to use ACLs in the FastIron X-Series devices.	N/A
Secure Shell (SSH) Version 2	In releases prior to 03.0.00, SSH version 1 (SSHv1) is supported. Starting with release 03.0.00, SSH version 2 (SSHv2) is supported and SSHv1 is no longer supported.	62
Restricting Telnet and SSH access based on a client's MAC address	You can restrict Telnet and SSH access to management functions on the Foundry device based on the MAC address of a connecting client.	34
Dynamic ARP inspection	Dynamic ARP Inspection (DAI) can prevent Man in the Middle (MiM) or ARP spoofing/poisoning attacks from malicious ARP packets, or misconfiguration of client IP addresses.	52
DHCP snooping	Dynamic Host Configuration Protocol (DHCP) snooping can ward off MiM attacks, stop unauthorized DHCP servers, and prevent errors due to user misconfiguration of DHCP servers.	57
IP Source Guard	IP Source Guard is used on client ports to prevent IP source address spoofing.	60
Secure copy	Secure Copy (SCP) uses security built into SSH to transfer image and configuration files to and from the device.	69

Enhancements to Username and Password	The following rules have been implemented to enhance the password features in the Foundry device. The following rules are enabled by default. • Users are now required to accept the message of the day. • Users are locked out (disabled) if they fail to login after three attempts. This feature is automatically enabled in release 03.0.00. You can use the **disable-on-login-failure** command to change the number of login attempts (up to 10) before users are locked out. The following rules are disabled by default: • Enhanced user password combination requirements • User password masking • Quarterly updates of user passwords • You can configure the system to store up to 15 previously configured passwords for each user. • You can use the **disable-on-login-failure** command to change the number of login attempts (up to 10) before users are locked out. • A password can now be set to expire.	74

System-Level Enhancements in 03.0.00

Specifying a minimum number of ports for a trunk group	You can configure the Foundry device to disable all of the ports in a trunk group when the number of active member ports drops below a specified threshold value.	35
Protected Link Groups	You can configure protected link groups to minimize disruption to the network by protecting critical links from loss of data and power.	35
ACL logging	You can enable the software to log entries in the Syslog for packets that are denied by ACL filters.	37
Using ACLs to filter ARP packets	You can use ACLs to filter ARP request packets.	50
Specifying which IP address will be included in a DHCP/ BOOTP reply packet.	You can configure the Foundry device so that a BOOTP/DHCP reply to a client contains the server's IP address as the source address instead of the router's IP address.	51
Port link dampening	Port link dampening allows you to configure a wait period before a port, whose link goes down then up, becomes enabled.	70
DNS list	DNS list support enables you to create a list of domain names that can be used to resolve host names. This list can have more than one domain name. When a client performs a DNS query, all hosts within the domains in the list can be recognized and queries can be sent to any domain on the list.	72

Cisco 7971 IP phone support	This release provides Voice VLAN support for Cisco's 7971 IP phones.	N/A
Ability to configure IPG	You can configure the Interpacket Gap (IPG), which is the time delay, in bit time, between frames transmitted by the device.	76
SNMP Version 3 traps	The Foundry device now supports SNMP notifications in SMIv2 format. This allows notifications to be encrypted and sent to the target hosts in a secure manner.	78
Displaying VLANs in alphanumeric order	In releases prior to 03.0.00, the output of some **show** commands list VLANs in the order they were configured. Starting with release 03.0.00, the VLANs are listed in alphanumeric order. This is the default behavior.	77
Disabling support for POE legacy devices	By default, the Foundry device automatically supports most POE legacy devices. Starting with release 03.0.00, if desired, you can disable support for POE legacy devices.	78
Lower fan operating noise	In release 03.0.00, the fans in the chassis operate at four speeds, at a reduced noise level. See the *Foundry FastIron X-Series Chassis Hardware Installation Guide* for details.	N/A

Feature Support

The FastIron X-Series devices support many of the applicable system-level, Layer 2 and Layer 3 features supported by the FastIron Edge Switch products and the BigIron Chassis devices. The FWSX supports Layer 2 features only.

The features that are available on your device depend on the type of software image the device is running. You can run one of the following types of software images:

- Layer 2 (supported on all FastIron X-Series devices)
- Base Layer 3 (not supported on FWSX models / supported on all other FastIron X-Series devices)
- Full Layer 3 (not supported on FWSX models / supported on all other, premium FastIron X-Series devices)

Table 2 lists the software that is loaded into the device's primary and secondary flash areas at the factory. All the flash images are included on the CD-ROM shipped with the device.

Table 2: Default Software Loads

Model[a]	**Software Images**	
	Primary Flash	**Secondary Flash**
FESX424 FESX424HF FESX424-POE FESX448 FSX FSX 800 FSX 1600	Layer 2	Base Layer 3

Table 2: Default Software Loads

Model[a]	Software Images	
	Primary Flash	**Secondary Flash**
FESX424-PREM FESX424HF-PREM FESX424-POE PREM FESX448 PREM FSX-PREM FSX 800-PREM FSX1600-PREM	Full Layer 3	Layer 2
FWSX424 FWSX448	Layer 2	N/A

[a] DC models included, where applicable.

Feature Highlights

Table 3 lists the primary features supported in this release.

Table 3: Feature Highlights

Category	Notes
Management Features Supported	
• Access Control Lists (ACLs) for controlling management access	
• IronView Network Manager (standalone and HP OpenView GUI)	Optional
• Serial and Telnet access to industry-standard Command Line Interface (CLI)	
• SNMP V1, V2, V3	
• Web-based GUI	
Security Features Supported	
• 802.1X port security	• Dynamic assignment for ACL, MAC filter, and VLAN • Multiple-host authentication
• Access Control Lists (ACLs) for filtering transit traffic	• Support for inbound ACLs. Outbound ACLs are not supported. • The maximum number of ACL filters (also called ACL entries) supported in the Layer 2 switch code for the entire switch is: • 1015 in releases 02.5.00 and earlier • 8192 starting in release 03.0.00 • The maximum number of ACL filters supported in the Layer 3 router code is: • 4096 in releases 2.5.00 and earlier • 8192 starting in release 03.0.00
• Address locking	
• Authentication, Authorization and Accounting (AAA)	RADIUS, TACACS/TACACS+
• Denial of Service (DoS) protection	TCP SYN Attacks and ICMP Attacks
• DHCP snooping	Support added in release 03.0.00
• Layer 2 MAC filtering	Filtering on source and destination MAC addresses supported
• Local passwords	
• MAC port security	
• Multi-device port authentication	

Table 3: Feature Highlights (Continued)

Category	Notes
• Secure Shell (SSH) version 2	Releases prior to 03.0.00 support SSH version 1.5 only.
• User accounts	
System Level Features Supported	
• 10/100/1000 port speed	
• 802.3ad dynamic link aggregation	• Up to 4-port trunk groups • Starting with release 03.0.00, 802.3ad is supported in the base Layer 3 code.
• ACL-based rate limiting	Support for ACL-based fixed and adaptive rate limiting on inbound ports
• ACL logging	• Support added in release 03.0.00 • Packets that are denied by ACL filters are logged in the Syslog
• ACL statistics	
• Auto MDI/MDIX	
• Broadcast, multicast, and unknown-unicast limiting	
• DiffServ support	
• Fixed Rate Limiting	• Port-based rate limiting on inbound ports • Fixed rate limiting is not supported on 10-Gigabit Ethernet ports. • Fixed rate limiting is not supported on tagged ports in the full Layer 3 router image (SXR0*xxxx*.bin).
• Foundry Discovery Protocol (FDP) / Cisco Discovery Protocol (CDP)	
• Jumbo frames	Up to 9216 bytes
• Multiple Syslog server logging	Up to six Syslog servers
• OSPF Version 2 MIB	RFC 1850
• Outbound rate shaping	Support added in release 03.0.00
• P-Bridge and Q-Bridge MIBs	RFC 2674
• Port monitoring and mirroring	
• Priority mapping using ACLs	
• Protected link groups	Support added in release 03.0.00
• sFlow	• For inbound traffic only • RFC 3176
• Static MAC entries with option to set priority	

Table 3: Feature Highlights (Continued)

Category	Notes
• Trunk groups	• The FESX424 and FWSX424 support up to 13 trunk groups containing 2, 3, or 4 ports. • The FESX448 and FWSX448 support up to 25 trunk groups containing 2, 3, or 4 ports. • The FSX, FSX8, and FSX16 support up to 31 trunk groups with 2, 3, or 4 ports.
Layer 2 Features Supported	
• 802.1d Spanning Tree Support	• PVST/PVST+ compatibility • PVRST compatibility (support added in release 02.5.00) • Enhanced IronSpan support includes Fast Port Span, and Single-instance Span • Rapid Spanning Tree support allows for sub-second convergence (draft 3 supported) • Foundry Layer 3 devices (routers) support up to 254 spanning tree instances for VLANs • Foundry Layer 2 devices (switches) support up to 255 spanning tree instances for VLANs
• 802.1p Quality of Service (QoS)	• Strict Priority (SP) • Weighted Round Robin (WRR) • Combined SP and WRR • 8 priority queues
• 802.1W Rapid Spanning Tree	
• Dynamic Host Configuration Protocol (DHCP) Assist	
• IGMPv2 snooping (Layer 2 Multicast)	
• Metro Ring Protocol 1 (MRP 1)	FESX and FSX devices can be MRP masters or MRP members (for different rings). In the FESX and FSX, the RHP received counter on non-master MRP nodes increment. This is different on other devices that support MRP 1.
• Metro Ring Protocol 2 (MRP 2)	Support added in release 03.0.00
• Topology groups	
• Uni-directional Link Detection (UDLD) (Link keepalive)	
• Virtual Cable Testing (VCT) Technology	You can diagnose a cable using Time Domain Reflectometry (TDR) technology
• Virtual Switch Redundancy Protocol (VSRP)	

Table 3: Feature Highlights (Continued)

Category	Notes
• VLAN Support	• 802.1Q with tagging • 802.1Q-in-Q Super Aggregated VLANs (SAVs) • Dual-mode VLANs • GVRP • Private VLANs (untagged ports only) • Protocol VLANs (IPv4, dynamic IPv6, IPX, and AppleTalk) • Layer 3 Subnet VLANs (IP subnet network, IPX, and AppleTalk) • Virtual routing interfaces • VLAN groups
• VSRP and MRP Signaling	Support added in 02.4.00
• VSRP Fast Start	Support added in 02.4.00
• Wire-speed Layer 2 switching	
Base Layer 3 Features Supported	
• IGMP V1, V2, and V3 (Layer 3 Multicast)	Support for IGMP V3 added in 02.4.00
• RIP V1 and V2	Static RIP support only. The Foundry device with base Layer 3 does not learn RIP routes from other Layer 3 devices. However, the device does advertise directly connected routes.
• Routing for directly connected IP subnets	
• Static IP	Up to 4000 static IP route entries
• Virtual Interfaces	Up to 512 virtual interfaces
• VRRP	Support for VRRP in base Layer 3 added in release 02.4.00
Full Layer 3 Features Supported	
• BGP4	• 4 BGP peers • Up to 100,000 BGP4 routes
• IGMP V1, V2, and V3	Support for IGMP V3 added in 02.4.00
• IP	
• IP Multicast	DVMRP, PIM Sparse (PIM SM), PIM Dense (PIM DM)
• OSPF	
• Policy-Based Routing (PBR)	Support added in 03.0.00
• RIP V1 and V2	

Table 3: Feature Highlights (Continued)

Category	Notes
• Route-only support	You can disable Layer 2 switching on a global basis as well as on an individual interface. **NOTE:** This feature is not supported on virtual interfaces.
• VRRP and VRRP-E	
• VRRP-E slow start timer	Support added in 03.0.00
• VSRP, VRRP and VRRP-E scale timer	Support added in 03.0.00

List of Unsupported Features

The FastIron X-Series devices do not support the following features. If required, these features are available on other Foundry devices.

Table 4: Unsupported Features

Category	Notes
System-Level Features not Supported	
• ACL logging is not supported for permitted packets.	ACL logging is supported for denied packets (denied packets are sent to the CPU for logging)
• Broadcast and multicast filters	
• NetFlow	
• Outbound ACLs	Inbound ACLs are supported
Layer 2 Features not Supported	
• IGMP V3 snooping	
• SuperSpan	
• VLAN-based priority	
Layer 3 Features not Supported	
• AppleTalk	
• Foundry Standby Router Protocol (FSRP)	
• IPX	
• IS-IS	
• Multiprotocol Border Gateway Protocol (MBGP)	
• Multiprotocol Label Switching (MPLS)	
• Multiprotocol Source Discovery Protocol (MSDP)	

Table 4: Unsupported Features (Continued)

Category	Notes
• Network Address Translation (NAT)	

Feature Documentation

For feature descriptions and configuration information, see the remaining sections in these release notes and the Foundry product manuals listed in “Where To Get More Information” on page 82.

Software Image Files

To use the features in this release, you need to run the software listed in Table 5.

NOTE: Your FastIron X-Series device must be running software release 02.2.00 or higher or 02.2.01a or higher, respectively, before upgrading to release 03.0.01. See “Migrating to the New Release” on page 13 for more information.

Table 5: Software Image Files

Device	Boot Image	Flash Image
FESX, FSX, FSX 800, and FSX 1600	SXZ03001.bin	SXS03001c.bin (Layer 2) SXL03001c.bin (Base Layer 3) SXR03001c.bin (Full Layer 3)
FWSX	FWXZ03001.bin	

The software is loaded at the factory. Table 2 on page 6 lists the default software loads for the device. All the software images are provided on the software CD-ROM shipped with the device.

NOTE: The software described in these notes applies only to the FastIron X-Series devices. You cannot use this software on the FES products, other Foundry Stackable devices, or on Foundry Chassis devices.

Upgrading Software

Use the following procedures to upgrade the software.

NOTE: This section does not describe how to upgrade a FastIron X-Series device to a premium (PREM) model. To perform this upgrade, you need an upgrade kit. Contact Foundry Networks for information.

Migrating to the New Release

Beginning with release 02.3.01, FESX and FSX devices share the same flash images. In releases prior to 02.3.01, FESX and FSX flash images were separate and were issued via separate software releases. Starting with release 02.3.01, the flash images for these devices were merged and are now issued in the same software release.

The new, combined flash images may create unique software upgrade circumstances for FESX and FSX devices. (FWSX devices are not affected by the software merge.) If your device is currently running software release 02.2.00 or later (FESX devices), or 02.2.01a or later (FSX devices), your device is not affected by the software merge. However, if your FESX or FSX device is running a release earlier than these versions, you must first upgrade the software on your device to FESX release 02.2.00 or later, or FSX release 02.2.01a or later, *before* loading the new software image. Earlier releases will not allow you to load the 02.3.01 or later software image.

To determine which software version is running on your device, use the **show version** command.

See the following sections for information on how to upgrade the software images on your device.

Upgrading from FESX pre-02.2.00 or FSX pre-02.2.01a to the New Release

If your device is running a software release earlier than FESX 02.2.00 or FSX 02.2.01a, you must first upgrade it to FESX 02.2.00 or later, or FSX 02.2.01a or later, before you can upgrade it to the new release. Follow the instructions, below.

1. Upgrade your device to software release FESX 02.2.00 or later, or FSX 02.2.01a or later. Follow the steps presented in "Upgrading the Boot Code" on page 14 and "Upgrading the Flash Code" on page 15. Make sure you reload the software after loading the flash code.
2. If you are upgrading a FESX from release 02.4.00 or earlier to release 03.0.01c, you must first upgrade the FESX to software release 02.5.00 before upgrading to release 03.0.01c. Note that this applies to the FESX only. All other X-Series devices are not affected. Follow the steps presented in "Upgrading the Boot Code" on page 14 and "Upgrading the Flash Code" on page 15. Make sure you reload the software after loading the flash code.
3. Upgrade your device to the new software release. Refer to one of the following sections:
   - FESX – "Upgrading from FESX 02.2.00 or later to the New Release" on page 14.
   - FSX – "Upgrading from FSX 02.2.01a or later to the New Release" on page 14.

Upgrading from FESX 02.2.00 or later to the New Release

1. If you are upgrading a FESX from release 02.4.00 or earlier to release 03.0.01c, you must first upgrade the FESX to software release 02.5.00 before upgrading to release 03.0.01c. Note that this applies to the FESX only. All other X-Series devices are not affected. Follow the steps presented in "Upgrading the Boot Code" on page 14 and "Upgrading the Flash Code" on page 15. Make sure you reload the software after loading the flash code.
2. Upgrade the boot code to the new version (SXZ0*xxxx*.bin) using the steps presented in "Upgrading the Boot Code" on page 14.
3. Upgrade the flash code to the new version using the steps presented in "Upgrading the Flash Code" on page 15.

Upgrading from FSX 02.2.01a or later to the New Release

1. Upgrade the boot code to the new version (SXZ0*xxxx*.bin) using the steps presented in "Upgrading the Boot Code" on page 14.
2. Upgrade the flash code to the new version using the steps presented in "Upgrading the Flash Code" on page 15.

Upgrading the Boot Code

NOTE: If you are upgrading a FESX or FSX device, see "Migrating to the New Release" on page 13 before performing the steps in this section.

1. Place the new boot code on a TFTP server to which the Foundry device has access.
2. Enter the following command at the Privileged EXEC level of the CLI (example: `FESX448 Switch#`) to copy the boot code from the TFTP server into flash memory:
   - **copy tftp flash** <ip-addr> <image-file-name> **bootrom**
3. Verify that the code has been successfully copied by entering the following command at any level of the CLI:
   - **show flash**

The output will display the compressed boot ROM code size and the boot code version.

4. Upgrade the flash code as instructed in the following section.

Upgrading the Flash Code

NOTE: If you are upgrading a FESX or FSX device, see "Migrating to the New Release" on page 13 before performing the steps in this section.

1. Place the new flash code on a TFTP server to which the Foundry device has access.
2. Enter the following command at the Privileged EXEC level of the CLI (example: `FESX448 Switch#`) to copy the flash code from the TFTP server into the flash memory:
    - **copy tftp flash** <ip-addr> <image-file-name> **primary | secondary**
3. Verify that the flash code has been successfully copied by entering the following command at any level of the CLI:
    - **show flash**
4. If the flash code version is correct, go to Step 5. Otherwise, go to Step 1.
5. Reload the software by entering one of the following commands:
    - **reload** (this command boots from the default boot source, which is the primary flash area by default)
    - **boot system flash primary | secondary**

Managing the Device

You can manage the Foundry device using any of the following applications:

- Command Line Interface (CLI) – a text-based interface accessible through a direct serial connection, a Telnet session, or an encrypted SSH session.
- Web management interface – A GUI-based management interface accessible through an HTTP (web browser) connection.
- IronView Network Manager – Optional standalone SNMP-based GUI application.

Logging on Through the CLI

After you configure an IP address, you can access the CLI either through the direct serial connection to the device or through a local or remote Telnet session.

You can initiate a local Telnet or SNMP connection by attaching a straight-through RJ-45 cable to a port and specifying the assigned management station IP address.

The commands in the CLI are organized into the following levels:

- User EXEC – Lets you display information and perform basic tasks such as pings and traceroutes.
- Privileged EXEC – Lets you use the same commands as those at the User EXEC level plus configuration commands that do not require saving the changes to the system-config file.
- CONFIG – Lets you make configuration changes to the device. To save the changes across reboots, you need to save them to the system-config file. The CONFIG level contains sub-levels for individual ports, for VLANs, for routing protocols, and other configuration areas.

NOTE: By default, any user who can open a serial or Telnet connection to the Foundry device can access all these CLI levels. To secure access, you can configure Enable passwords or local user accounts, or you can configure the device to use a RADIUS or TACACS/TACACS+ server for authentication. See the *Foundry Security Guide* for more information.

On-Line Help

To display a list of available commands or command options, enter “?” or press Tab. If you have not entered part of a command at the command prompt, all the commands supported at the current CLI level are listed. If you enter part of a command, then enter “?” or press Tab, the CLI lists the options you can enter at this point in the command string.

If you enter an invalid command followed by ?, a message appears indicating the command was unrecognized. For example:

```
FESX424 Switch(config)# rooter ip ?
Unrecognized command
```

Command Completion

The CLI supports command completion, so you do not need to enter the entire name of a command or option. As long as you enter enough characters of the command or option name to avoid ambiguity with other commands or options, the CLI understands what you are typing.

Scroll Control

By default, the CLI uses a page mode to paginate displays that are longer than the number of rows in your terminal emulation window. For example, if you display a list of all the commands at the global CONFIG level but your terminal emulation window does not have enough rows to display them all at once, the page mode stops the display and lists your choices for continuing the display.

Here is an example:

```
aaa
all-client
arp
boot
```

some lines omitted for brevity...

```
logging
mac
--More--, next page: Space, next line:
Return key, quit: Control-c
```

The software provides the following scrolling options:

- Press the Space bar to display the next page (one screen at time).
- Press the Return or Enter key to display the next line (one line at a time).
- Press CTRL + C to cancel the display.

Line Editing Commands

The CLI supports the following line editing commands. To enter a line-editing command, use the CTRL-key combination for the command by pressing and holding the CTRL key, then pressing the letter associated with the command.

Table 6: CLI Line Editing Commands

Ctrl-Key Combination	Description
Ctrl-A	Moves to the first character on the command line.
Ctrl-B	Moves the cursor back one character.
Ctrl-C	Escapes and terminates command prompts and ongoing tasks (such as lengthy displays), and displays a fresh command prompt.
Ctrl-D	Deletes the character at the cursor.

Table 6: CLI Line Editing Commands (Continued)

Ctrl-Key Combination	Description
Ctrl-E	Moves to the end of the current command line.
Ctrl-F	Moves the cursor forward one character.
Ctrl-K	Deletes all characters from the cursor to the end of the command line.
Ctrl-L; Ctrl-R	Repeats the current command line on a new line.
Ctrl-N	Enters the next command line in the history buffer.
Ctrl-P	Enters the previous command line in the history buffer.
Ctrl-U; Ctrl-X	Deletes all characters from the cursor to the beginning of the command line.
Ctrl-W	Deletes the last word you typed.
Ctrl-Z	Moves from any CONFIG level of the CLI to the Privileged EXEC level; at the Privileged EXEC level, moves to the User EXEC level.

For a complete list of CLI commands and syntax information for each command, see the *Foundry Switch and Router Command Line Interface Reference*.

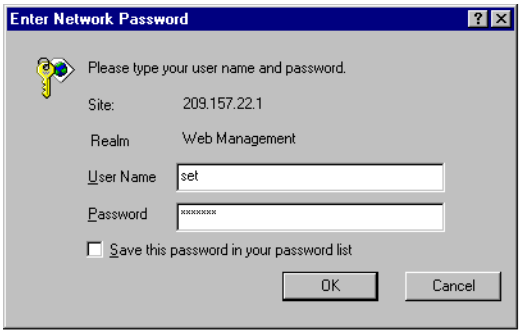
Logging On Through the Web Management Interface

To use the Web management interface, open a web browser and enter the IP address of the Foundry device in the Location or Address field. The Web browser contacts the Foundry device and displays a login panel.

NOTE: If you are unable to connect with the device through a Web browser due to a proxy problem, it may be necessary to set your Web browser to direct Internet access instead of using a proxy. For information on how to change a proxy setting, refer to the on-line help provided with your Web browser.

To log in, click on the Login link. The following dialog displays.

Figure 1 Web management interface login dialog



Enter Network Password

Please type your user name and password.

Site: 209.157.22.1

Realm Web Management

User Name set

Password *******

□ Save this password in your password list

By default, you can use the user name “get” and the default read-only password “public” for read-only access. However, for read-write access, you must enter “set” for the user name, and enter a read-write community string you have configured on the device for the password.

There is no default read-write community string. You must add one using the CLI. To add an encrypted community string, enter a command such as the following:

```
FESX448 Switch(config)# snmp-server community private rw
```

Syntax: snmp-server community <string> ro | rw

The <string> parameter specifies the community string name. The string can be up to 32 characters long.

The **ro | rw** parameter specifies whether the string is read-only (**ro**) or read-write (**rw**).

As an alternative to using the SNMP community strings to log in, you can configure the Foundry device to secure Web management access using local user accounts or Access Control Lists (ACLs). See the *Foundry Security Guide*.

Navigating the Web Management Interface

When you log into a device, the System configuration panel is displayed. This panel allows you to enable or disable major system features. You can return to this panel from any other panel by selecting the Home link.

The Site Map link gives you a view of all available options on a single screen.

The left pane of the Web management interface window contains a “tree view,” similar to the one found in Windows Explorer. Configuration options are grouped into folders in the tree view. These folders, when expanded, reveal additional options. To expand a folder, click on the plus sign to the left of the folder icon.

You can change the appearance of the Web management interface by using one of the following methods.

USING THE CLI

Using the CLI, you can modify the appearance of the Web management interface with the **web-management** command.

To cause the Web management interface to display the List view by default:

```
FESX424 Switch(config)# web-management list-menu
```

To disable the front panel frame:

```
FESX424 Switch(config)# no web-management front-panel
```

When you save the configuration with the **write memory** command, the changes will take place the next time you start the Web management interface, or if you are currently running the Web management interface, the changes will take place when you click the Refresh button on your browser.

USING THE WEB MANAGEMENT INTERFACE

1. Click on the plus sign next to Configure in the tree view to expand the list of configuration options.
2. Click on the plus sign next to System in the tree view to expand the list of system configuration links.
3. Click on the plus sign next to Management in the tree view to expand the list of system management links.
4. Click on the Web Preference link to display the Web Management Preferences panel.
5. Enable or disable elements on the Web management interface by clicking on the appropriate radio buttons on the panel. The following figure identifies the elements you can change.

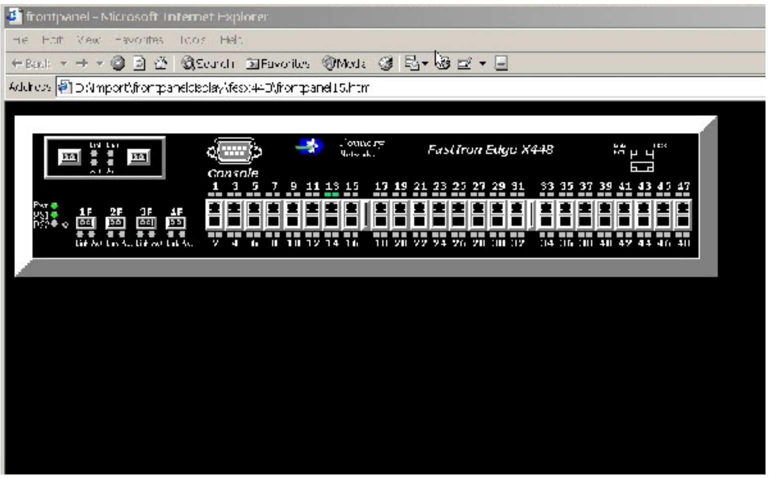
NOTE: The tree view is available when you use the Web management interface with Netscape 4.0 or higher or Internet Explorer 4.0 or higher browsers. If you use the Web management interface with an older browser, the Web management interface displays the List view only, and the Web Management Preferences panel does not include an option to display the tree view.

6. When you have finished, click the Apply button on the panel, then click the Refresh button on your browser to activate the changes.
7. To save the configuration, click the plus sign next to the Command folder, then click the Save to Flash link.

NOTE: The only changes that become permanent are the settings to the Menu Type and the Front Panel Frame. Any other elements you enable or disable will go back to their default settings the next time you start the Web management interface.

You can view a picture of the device's front panel, as shown in Figure 2, by clicking on the Monitor -> Front Panel menu option.

Figure 2 **FastIron Edge Switch X-Series front panel in Web management interface**



Recovering from a Lost Password

By default, the CLI does not require passwords. However, if someone has configured a password for the device but the password has been lost, you can regain super-user access to the device using the following procedure.

NOTE: Recovery from a lost password requires direct access to the serial port and a system reset.

To recover from a lost password:

1. Start a CLI session over the serial interface to the Foundry device.
2. Reboot the device.
3. While the system is booting, before the initial system prompt appears, enter **b** to enter the boot monitor mode.
4. Enter **no password** at the prompt. (You cannot abbreviate this command.)
5. Enter **boot system flash primary** at the prompt. This command causes the device to bypass the system password check.
6. When the console prompt reappears, you will be prompted to enter a new login name and password. For example:

   ```
   Please Enter Login Name:
   Please Enter Password:
   ```

   Enter the new login name and password.

Displaying and Saving Configuration Changes

When you make a configuration change, the change enters the device's running configuration but is not saved if you reload the software. To make a change permanent, you must save the change to the device's startup-config file.

Displaying Configuration Changes

To display the running configuration, enter the following command from any level of the CLI:

```
FESX424 Switch# show running-config
```

To display the startup configuration, enter the following command from any level of the CLI:

```
FESX424 Switch# show configuration
```

NOTE: You cannot display the running-config or startup-config file using the Web management interface.

Saving Configuration Changes

To permanently save a configuration change so that the change stays in effect following a software reload, use one of the following methods.

USING THE CLI

To replace the startup configuration with the running configuration, enter the following command at any Privileged EXEC or CONFIG command prompt:

```
FESX424 Switch# write memory
```

USING THE WEB MANAGEMENT INTERFACE

1. Click on the plus sign next to Command in the tree view to expand the list of command options.
2. Select the Save to Flash option.
3. Select Yes when the Web management interface asks you whether you really want to save the configuration changes to flash.

Addenda to the Configuration Guide

This section provides clarification to the information in the *Foundry FastIron X-Series Configuration Guide*.

Using an Extended ACL as Input to the Distribution List

NOTE: This section describes the procedure of using an Extended ACL as input to the OSPF distribution list.This is an addendum to the *Foundry FastIron X-Series Configuration Guide*. This is not a new feature

You can use an extended ACL with an OSPF distribution list to filter OSPF routes based on the network mask of the destination network.

To use an extended ACL to configure an OSPF distribution list for denying specific routes, enter commands such as the following:

```
FESX424 Router(config)# ip access-list extended no_ip
FESX424 Router(config-ext-nacl)# deny ip 4.0.0.0 0.255.255.255 255.255.0.0
0.0.255.255
FESX424 Router(config-ext-nacl)# permit ip any any
FESX424 Router(config-ext-nacl)# exit
FESX424 Router(config)# router ospf
FESX424 Router(config-ospf-router)# distribute-list no_ip in
```

The first three commands configure an extended ACL that denies routes to any 4.x.x.x destination network with a 255.255.0.0 network mask and allows all other routes for eligibility to be installed in the IP route table. The last three commands change the CLI to the OSPF configuration level and configure an OSPF distribution list that uses the ACL as input. The distribution list prevents routes to any 4.x.x.x destination network with network mask 255.255.0.0 from entering the IP route table. The distribution list does not prevent the routes from entering the OSPF database.

Syntax: [no] ip access-list extended <acl-name> | <acl-id>

Syntax: deny | permit <ip-protocol> <source-ip> <wildcard> <destination-ip> <wildcard>

The <acl-name> | <acl-id> parameter specifies the ACL name or ID.

The **deny | permit** parameter indicates whether packets that match the policy are dropped or forwarded.

The <ip-protocol> parameter indicates the type of IP packet you are filtering. When using an extended ACL as input for an OSPF distribution list, specify **ip**.

Since this ACL is input to an OSPF distribution list, the <source-ip> parameter actually specifies the destination network of the route.

The <wildcard> parameter specifies the portion of the source address to match against. The <wildcard> is a four-part value in dotted-decimal notation (IP address format) consisting of ones and zeros. Zeros in the mask mean the packet's source address must match the <source-ip>. Ones mean any value matches. For example, the <source-ip> and <wildcard> values 4.0.0.0 0.255.255.255 mean that all 4.x.x.x networks match the ACL.

If you want the policy to match on all network addresses, enter **any any**.

If you prefer to specify the wildcard (mask value) in Classless Interdomain Routing (CIDR) format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of "4.0.0.0 0.255.255.255" as "4.0.0.0/8". The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into zeros.

NOTE: If you enable the software to display IP sub-net masks in CIDR format, the mask is saved in the file in "/<mask-bits>" format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the ACL entry regardless of whether the software is configured to display the masks in CIDR format.

If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with sub-net mask in the display produced by the **show ip access-list** commands.

Since this ACL is input to an OSPF distribution list, the <destination-ip> parameter actually specifies the subnet mask of the route.

The <wildcard> parameter specifies the portion of the subnet mask to match against. For example, the <destination-ip> and <wildcard> values 255.255.255.255 0.0.0.255 mean that subnet mask /24 and longer match the ACL.

If you want the policy to match on all network masks, enter **any any**.

Load Balancing Differences

NOTE: This section describes how the FastIron X Series load balances unknown unicast, multicast, and broadcast traffic. This is an addendum to the *Foundry FastIron X-Series Configuration Guide*. This is not a new feature.

The Foundry device load balances unknown unicast, multicast, and broadcast traffic based on the source port and VLAN ID and not on any source or destination information in the packet.

For example, when the switch receives unknown unicast, multicast, and broadcast packets, and the packets are from the same source port, the packets are forwarded to the same port of the trunk group. Conversely, when the switch receives unknown unicast, multicast, and broadcast packets, and the packets are from different source ports, the packets are load-balanced across all the ports of the trunk group.

Note that this does not apply to known unicast traffic, which is always load balanced across all the ports of a trunk group based on the traffic's Layer 2 and Layer 3 source and destination parameters.

Enhancement in 03.0.01a

This section describes the enhancement in release 03.0.01a.

Message of the Day

In previous releases, users were required to press the Enter key after the Message of the Day (MOTD) was displayed, prior to logging in to the Foundry device on a console or via a Telnet session. Beginning with this release, this requirement is disabled by default. Unless configured, users do not have to press Enter after the MOTD banner is displayed.

For example, if the MOTD "Authorized Access Only" is configured, by default, the following messages are displayed when a user access the Foundry device via Telnet:

```
Authorized Access Only ...

Username:
```

The user can then login to the device.

However, if the requirement to press the Enter key is enabled, the following messages are displayed when accessing the switch via Telnet:

```
Authorized Access Only ...

Press <Enter> to accept and continue the login process....
```

The user must press the Enter key before the login prompt is displayed.

Also, on the console, the following messages are displayed if the requirement to press the Enter key is disabled:

```
Press Enter key to login

Authorized Access Only ...

User Access Verification

Please Enter Login Name:
```

However, if the requirement to press the Enter key after a MOTD is enabled, the following messages are displayed when accessing the switch on the console:

```
Press Enter key to login

Authorized Access Only ...

Press <Enter> to accept and continue the login process....
```

The user must press the Enter key to continue to the login prompt.

To enable the requirement to press the Enter key after the MOTD is displayed, enter a command such as the following:

```
FastIron SuperX Switch(config)#banner motd require-enter-key
```

Syntax: [no] banner motd require-enter-key

Use the **no** form of the command to disable the requirement.

Enhancements in 03.0.01

This section describes the software enhancements in release 03.0.01.

Specifying the Maximum Number of Entries Allowed in the RMON Control Table

Starting in software release 03.0.01, you can specify the maximum number of entries allowed in the RMON control table, including alarms, history, and events. In addition, the default number of RMON entries allowed in the RMON control table has increased from 240 to 1024 on the FESX, and 2048 on the FSX, FSX 800, and FSX 1600. The maximum number of RMON entries supported is 32768.

For example, to set the maximum number of allowable entries to 3000 in the RMON history table, enter commands such as the following:

```
FastIron SuperX Switch(config)# system-max rmon-entries 3000
FastIron SuperX Switch(config)# write mem
FastIron SuperX Switch(config)# exit
FastIron SuperX Switch# reload
```

NOTE: You must save the change to the startup-config file and reload or reboot. The change does not take effect until your reload or reboot.

Syntax: system-max rmon-entries <value>

where <value> can be:

- 1536 – 32768 for FSX, FSX 800, and FSX 1600 devices
- 128 – 32768 for FESX devices

Enhancements in 03.0.00

This section describes the new software enhancements in release 03.0.00.

Additional IDs for MRP Rings

You can configure an MRP ring ID to be a number between 1 and 1023, using the following command:

```
FESX424 Router(config)# vlan 2
FESX424 Router(config-vlan-2)# metro-ring 1
```

Syntax: [no] metro-ring <ring-id>

Enter 1-1023 for <ring-id>; ID 256 is reserved for VSRP.

Metro Ring Protocol Phase 2

Metro Ring Protocol (MRP) Phase 2 expands the functionality of MRP by allowing the same physical interface to be shared by multiple rings belonging to the same VLAN.

MRP is a Foundry proprietary protocol that prevents Layer 2 loops and provides fast reconvergence in Layer 2 ring topologies. It is an alternative to STP and is especially useful in Metropolitan Area Networks (MANs) that require more nodes and faster reconvergence time than what STP provides.

NOTE: You can configure MRP Phase 2 with the SXR (Layer 3 switch code), SXS (Layer 2 switch code), and SXL (Layer 2 Switch code with base Layer 3 support) code streams.

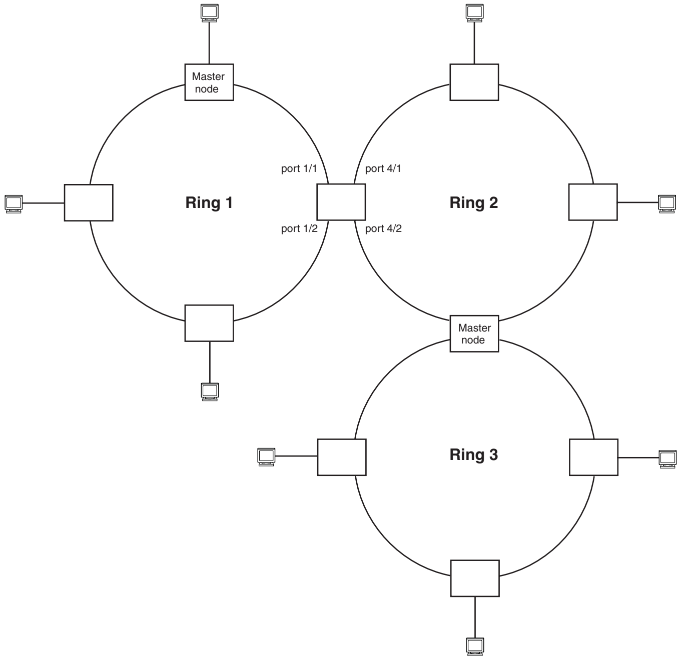
An MRP ring consists of nodes and each node has two interfaces on the ring. The interfaces on the ring must be members of the same port-based VLAN. Each node on the ring is connected to a separate customer network. The nodes forward Layer 2 traffic to and from the customer networks through the ring.

One node, is configured as the master node of the MRP ring. One of the two interfaces on the master node is configured as the primary interface; the other is the secondary interface. The primary interface originates Ring

Health Packets (RHPs), which are used to monitor the health of the ring. An RHP is forwarded on the ring to the next interface until it reaches the secondary interface of the master node. The secondary interface blocks the packet to prevent a Layer 2 loops.

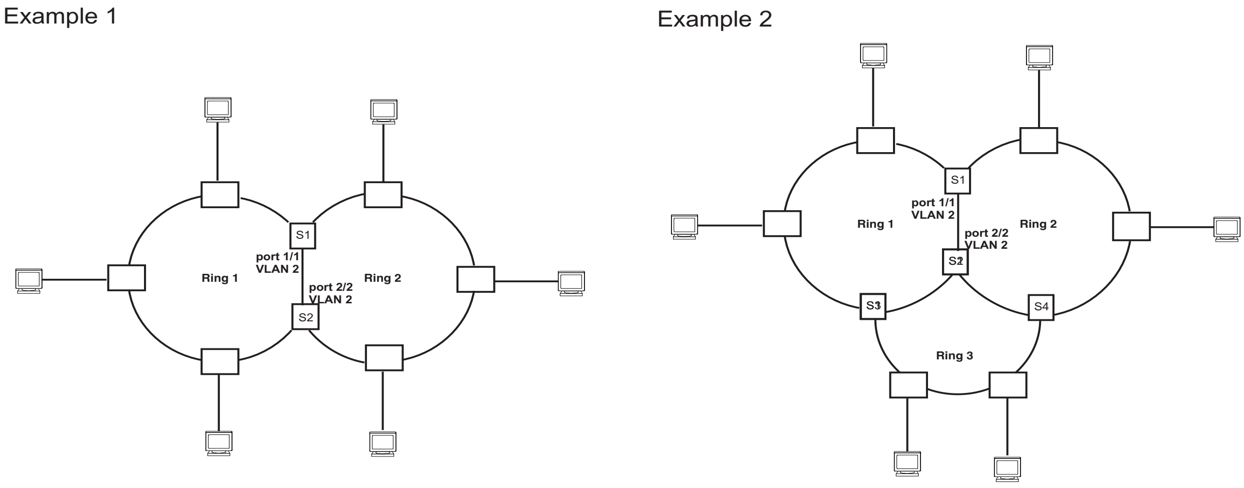
In MRP Phase 1, a node can have multiple MRP rings, but the rings cannot share the same interface. Also, when you configured an MRP ring, any node on the ring that is a BigIron Chassis device can be designated as the master node for the ring. A master node can be the master node of more than one ring. (See Figure 3.) Each ring is an independent ring and RHP packets are processed within each ring.

Figure 3 Multiple MRP Rings - MRP Phase 1



With MRP Phase 2, MRP rings can be configured to share the same interfaces as long as the interfaces belong to the same VLAN. Figure 4 shows examples of multiple MRP rings that share the same interface.

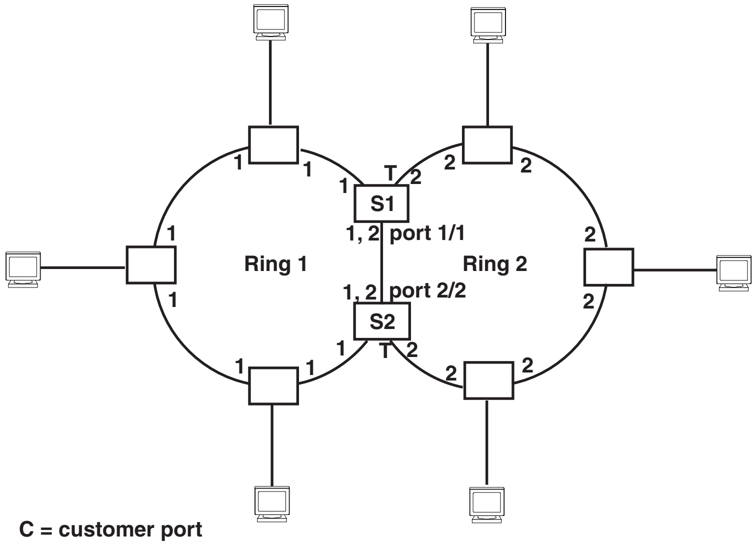
Figure 4 Examples of multiple rings sharing the same interface - MRP Phase 2



On each node that will participate in the ring, you specify the ring’s ID and the interfaces that will be used for ring traffic. In a multiple ring configuration, a ring’s ID determines its priority. The lower the ring ID, the higher priority of a ring.

A ring's ID is also used to identify the interfaces that belong to a ring.

Figure 5 Interface IDs and Types



C = customer port

For example, in Figure 5, the ID of all interfaces on all nodes on Ring 1 is 1 and all interfaces on all nodes on Ring 2 is 2. Port 1/1 on node S1 and Port 2/2 on S2 have the IDs of 1 and 2 since the interfaces are shared by Rings 1 and 2.

The ring's ID is also used to determine an interface's priority. Generally, a ring's ID is also the ring's priority and the priority of all interfaces on that ring. However, if the interface is shared by two or more rings, then the highest priority (lowest ID) becomes the priority of the interface. For example, in Figure 5, all interfaces on Ring 1, except for Port 1/1 on node S1 and Port 2/2 on node S2 have a priority of 1. Likewise, all interfaces on Ring 2, except for Port 1/1 on node S1 and Port 2/2 on node S2 have a priority of 2. Port 1/1 on S1 and Port 2/2 on S2 have a priority of 1 since 1 is the highest priority (lowest ID) of the rings that share the interface.

If a node has interfaces that have different IDs, the interfaces that belong to the ring with the highest priority become regular ports. Those interfaces that do not belong to the ring with the highest priority become tunnel ports. In Figure 5, nodes S1 and S2 have interfaces that belong to Rings 1 and 2. Those interfaces with a priority of 1 are regular ports. The interfaces with a priority of 2 are the tunnel ports since they belong to Ring 2, which has a lower priority than Ring 1.

Selection of Master Node

Allowing MRP rings to share interfaces limits the nodes that can be designated as the master node. Any node on an MRP ring that does not have a shared interface can be designated as the ring's master node. However, if all nodes on the ring have shared interfaces, nodes that do not have tunnel ports can be designated as the master node of that ring. If none of the nodes meet these criteria, you must change the rings' priorities by reconfiguring the rings' ID.

In Figure 5, any of the nodes on Ring 1, even S1 or S2, can be a master node since none of its interfaces are tunnel ports. However in Ring 2, neither S1 nor S2 can be a master node since these nodes contain tunnel ports.

RHP Processing in Rings with Shared Interfaces

Interfaces on an MRP ring have one of the following states:

- Preforwarding (PF) – All ring interfaces are in this state when you enable MRP.
- Forwarding (F) – An interface changes from Preforwarding to Forwarding when the port's preforwarding time expires.
- Blocking (B) – The interface cannot forward data. Only the secondary interface on the Master node can be Blocking.

The primary interface of the master node initiates the RHP packets and sends it on the ring. When the packet reaches an interface, MRP checks to see if the receiving interface is a regular port or a tunnel port:

If the port is a regular port, the RHP packet is forwarded to the next interface. Forwarding of the packet continues on the ring until the secondary interface of the master node receives the packet and blocks it.

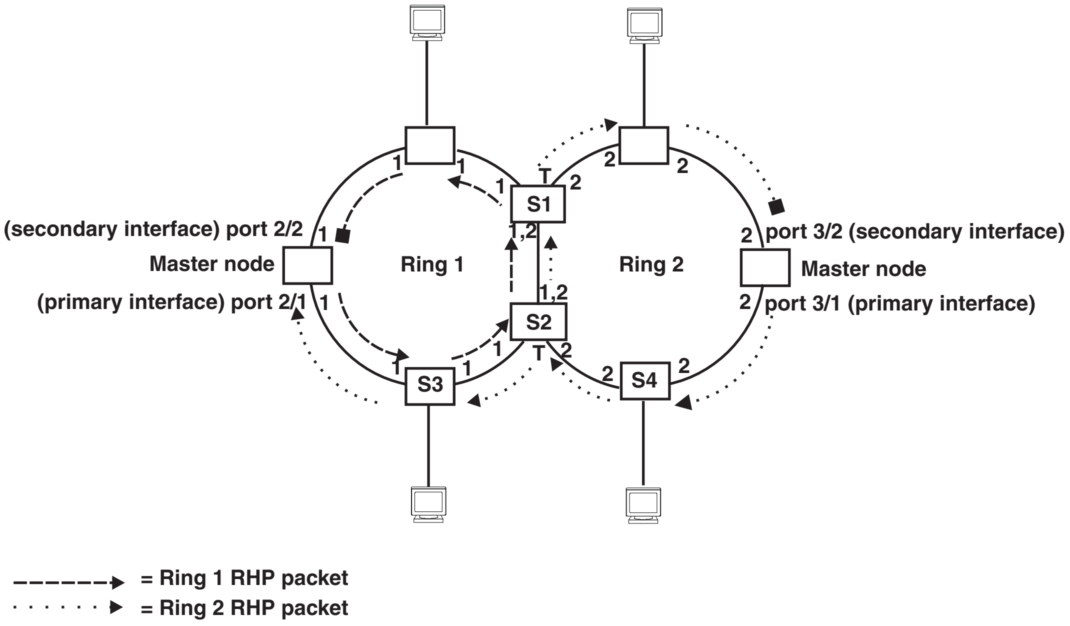
If the port is a tunnel port, MRP checks the priority of the RHP packet and compares it to the priority of the tunnel port.

- If the RHP packet's priority is less than or equal to the interface's priority, the packet is forwarded through that interface.
- If the priority of the RHP packet is greater than the priority of the interface, the RHP packet is dropped. MRP then assumes that the ring is broken since the secondary interface will not receive the packet.

Normal Flow

Figure 6 shows an example of how RHP packets are processed normally in MRP rings with shared interfaces.

Figure 6 Flow of RHP packets on MRP rings with shared interfaces


(secondary interface) port 2/2
Master node
(primary interface) port 2/1
Ring 1
Ring 2
S1
S2
S3
S4
T
T
1,2
1,2
1
2
port 3/2 (secondary interface)
Master node
port 3/1 (primary interface)
= Ring 1 RHP packet
= Ring 2 RHP packet

Port 2/1 on Ring 1's master node is the primary interface of the master node. The primary interface forwards an RHP packet on the ring. Since all the interfaces on Ring 1 are regular ports, the RHP packet is forwarded to all the interfaces until it reaches Port 2/2, the secondary interface of the master node. Port 2/2 then blocks the packet to complete the process.

On Ring 2, Port 3/1, is the primary interface of the master node. It sends an RHP packet on the ring. Since all ports on S4 are regular ports, the RHP packet is forwarded on those interfaces. When the packet reaches S2, the receiving interface is a tunnel port. The port compares the packet's priority to its priority. Since the packet's priority is the same as the tunnel port's priority, the packet is forwarded up the link shared by Rings 1 and 2.

When the RHP packet reaches the interface on node S2 shared by Rings 1 and 2, the packet is forwarded since its priority is less than the interface's priority. The packet continues to be forwarded to node S1 until it reaches the tunnel port on S1. That tunnel port determines that the RHP packet's priority is equal to the port's priority and forwards the packet. The RHP packet is forwarded to the remaining interfaces on Ring 2 until it reaches port 3/2, the secondary interface of the master node. Port 3/2 then blocks the packet to prevent a loop.

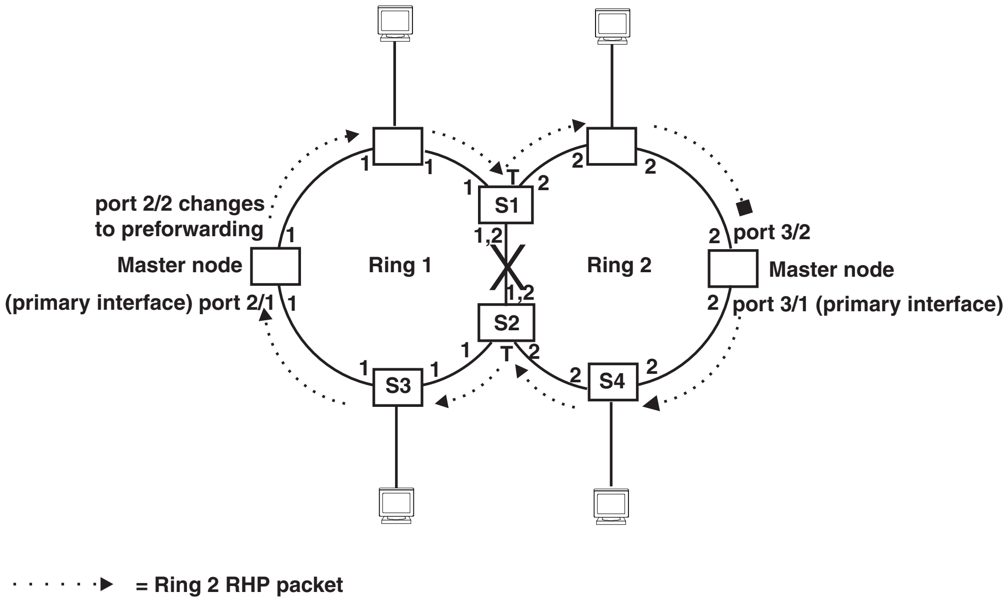
When the RHP packet from Ring 2 reached S2, it was also forwarded from S2 to S3 on Ring 1 since the port on S2 has a higher priority than the RHP packet. The packets is forwarded around Ring 1 until it reaches port 2/2, Ring 1's the secondary port. The RHP packet is then blocked by that port.

Flow When a Link Breaks

If the link between shared interfaces breaks (Figure 7), the secondary interface on Ring 1's master node changes to a preforwarding state. The RHP packet sent by port 3/1 on Ring 2 is forwarded through the interfaces on S4, then to S2. The packet is then forwarded through S2 to S3, but not from S2 to S1 since the link between the two nodes is not available. When the packet reaches Ring 1's master node, the packet is forwarded through the secondary interface since it is currently in a preforwarding state. A secondary interface in preforwarding mode ignores any RHP packet that is not from its ring. The secondary interface changes to blocking mode only when the RHP packet forwarded by its primary interface is returned.

The packet then continues around Ring 1, through the interfaces on S1 to Ring 2 until it reaches Ring 2's master node. Port 3/2, the secondary interface on Ring 2 changes to blocking mode since it received its own packet, then blocks the packet to prevent a loop.

Figure 7 Flow of RHP packets when a link for shared interfaces brakes



RHP packets follow this flow until the link is restored; then the RHP packet returns to it normal flow as shown in Figure 6.

Configuring MRP with Shared Interfaces

There are no new commands or parameters used to configure MRP with shared interfaces. However, the CLI now allows you to enter commands such as the following when configuring MRP:

```
FastIron SuperX Router(config)# vlan 2
FastIron SuperX Router(config-vlan-2)# metro-ring 1
FastIron SuperX Router(config-vlan-2-mrp-1)# name CustomerA
FastIron SuperX Router(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/
2
FastIron SuperX Router(config-vlan-2-mrp-1)# enable
FastIron SuperX Router(config-vlan-2-mrp-1)# metro-ring 2
FastIron SuperX Router(config-vlan-2-mrp-2)# name CustomerB
FastIron SuperX Router(config-vlan-2-mrp-2)# ring-interface ethernet 1/1 ethernet 1/
2
FastIron SuperX Router(config-vlan-2-mrp-1)# enable
```

Syntax: [no] metro-ring <ring-id>

The <ring-id> parameter specifies the ring ID. In releases prior to 03.0.00, the ring ID can be a value from 1 – 255. In release 03.0.00 and later, the ring ID can be a value from 1 – 1023; ID 256 is reserved for VSRP. Configure the same ring ID on each of the nodes in the ring.

Syntax: [no] name <string>

The <string> parameter specifies a name for the ring. The name is optional, but it can have up to 20 characters long and can include blank spaces. If you use a name that has blank spaces, enclose the name in double quotation marks (for example: "Customer A").

Syntax: [no] ring-interface ethernet <primary-if> ethernet <secondary-if>

The **ethernet** <primary-if> parameter specifies the primary interface. On the master node, the primary interface is the one that originates RHPs. Ring control traffic and Layer 2 data traffic will flow in the outward direction from this interface by default. On member nodes, the direction of traffic flow depends on the traffic direction selected by the master node. Therefore, on a member node, the order in which you enter the interfaces does not matter.

The **ethernet** <secondary-if> parameter specifies the secondary interface.

Syntax: [no] enable

The **enable** command enables the ring.

Displaying Metro Ring Information

There are no new commands to display metro ring information; however, the **show metro** command now includes Interface Type that indicates if an interface is a regular port or a tunnel port. To display ring information, enter the following command:

```
FastIron SuperX Router(config)# show metro

Metro Ring 2
=============
Ring       State     Ring       Master     Topo       Hello      Prefwing
id                   role       vlan       group      time(ms)   time(ms)
2          enabled   member     2          not conf   100        300

Ring interfaces    Interface role     Forwarding state    Active interface
Interface Type
ethernet 1/1      primary            disabled            none                Regular
ethernet 1/2       secondary          forwarding          ethernet 2          Tunnel

RHPs sent          RHPs rcvd          TC RHPs rcvd        State changes
```

Syntax: show metro [<ring-id>]

This display shows the following information.

Table 7: CLI Display of MRP Ring Information

This Field...	Displays...
Ring id	The ring ID
State	The state of MRP. The state can be one of the following: • enabled – MRP is enabled • disabled – MRP is disabled

Table 7: CLI Display of MRP Ring Information (Continued)

This Field...	Displays...
Ring role	Whether this node is the master for the ring. The role can be one of the following: • master • member
Master vlan	The ID of the master VLAN in the topology group used by this ring. If a topology group is used by MRP, the master VLAN controls the MRP settings for all VLANs in the topology group. **Note**: The topology group ID is 0 if the MRP VLAN is not the master VLAN in a topology group. Using a topology group for MRP configuration is optional.
Topo group	The topology group ID.
Hello time	The interval, in milliseconds, at which the Forwarding port on the ring's master node sends Ring Hello Packets (RHPs).
Prefwing time	The number of milliseconds an MRP interface that has entered the Preforwarding state will wait before changing to the Forwarding state. If a member port in the Preforwarding state does not receive an RHP within the Preforwarding time (Prefwing time), the port assumes that a topology change has occurred and changes to the Forwarding state. The secondary port on the Master node changes to Blocking if it receives an RHP, but changes to Forwarding if the port does not receive an RHP before the preforwarding time expires. **Note**: A member node's Preforwarding interface also changes from Preforwarding to Forwarding if it receives an RHP whose forwarding bit is on.
Ring interfaces	The device's two interfaces with the ring. **Note**: If the interfaces are trunk groups, only the primary ports of the groups are listed.
Interface role	The interface role can be one of the following: • primary • Master node – The interface generates RHPs. • Member node – The interface forwards RHPs received on the other interface (the secondary interface). • secondary – The interface does not generate RHPs. • Master node – The interface listens for RHPs. • Member node – The interface receives RHPs.

Table 7: CLI Display of MRP Ring Information (Continued)

This Field...	Displays...
Forwarding state	Whether MRP Forwarding is enabled on the interface. The forwarding state can be one of the following: • blocking – The interface is blocking Layer 2 data traffic and RHPs • disabled – The interface is down • forwarding – The interface is forwarding Layer 2 data traffic and RHPs • preforwarding – The interface is listening for RHPs but is blocking Layer 2 data traffic
Active interface	The physical interfaces that are sending and receiving RHPs. **Note**: If a port is disabled, its state is shown as “disabled”. **Note**: If an interface is a trunk group, only the primary port of the group is listed.
RHPs sent	The number of RHPs sent on the interface. **Note**: This field applies only to the master node. On non-master nodes, this field contains 0. This is because the RHPs are forwarded in hardware on the non-master nodes.
RHPs rcvd	The number of RHPs received on the interface. **Note**: This field applies only to the master node. On non-master nodes, this field contains 0. This is because the RHPs are forwarded in hardware on the non-master nodes.
TC RHPs rcvd	The number of Topology Change RHPs received on the interface. A Topology Change RHP indicates that the ring topology has changed.
State changes	The number of MRP interface state changes that have occurred. The state can be one of the states listed in the Forwarding state field.
Interface Type	Shows if the interface is a regular port or a tunnel port.

Changes to Spanning Tree Port Priority

In releases prior to 03.0.00, when configuring spanning tree port priority, the valid range of values is 8 – 252, configurable in increments of four. The default is 128.

Starting with release 03.0.00, the spanning tree port priority configuration is changed such that the valid range of values is 0 – 240, configurable in increments of 16. The default is 128. This is in compliance with the IEEE 802.1w standards. The change is necessary to accommodate the increased number of ports in the FSX 1600 chassis, since the STP port-identifier field uses 12 bits for the port-number and restricts 4 bits for the port-priority.

To configure spanning tree port priority, enter a command such as the following at the Global CONFIG level of the CLI:

```
FastIron SuperX Switch(config)# spanning-tree ethernet 2/5 priority 16
```

Syntax: spanning-tree ethernet [<slot-num>/]<port-num> priority <value>

The <slot-num> parameter applies to chassis devices only.

For <port-num>, enter a valid port number.

The <value> parameter specifies the preference that STP gives this port relative to other ports for forwarding traffic out of the spanning tree. You can specify a value from 0 – 240, in increments of 16. If you enter a value that is not divisible by 16, the software returns an error message. The default value is 128. A higher numerical value means a lower priority; thus, the highest priority is 0.

Outbound Rate Shaping

Outbound Rate Shaping is a port level feature that is used to shape the rate and to control the bandwidth of outbound traffic on a port. This feature smooths out excess and bursty traffic to the configured maximum limit before it is sent out on a port. Packets are stored in available buffers and then forwarded at a rate no greater than the configured limit. This process provides for better control over the inbound traffic of neighboring devices.

The device has one global rate shaper for a port and one rate shaper for each port priority queue. Rate shaping is done on a single-token basis, where each token is defined to be 1 byte.

The following rules apply when configuring outbound rate shapers:

- Outbound rate shapers can be configured *only* on physical ports, not on virtual or loopback ports.
- For trunk ports, the rate shaper must be configured on individual ports of a trunk using the **config-trunk-ind** command (trunk configuration level); you cannot configure a rate shaper for a trunk.
- You can configure a rate shaper for a port and for the individual priority queues of that port. However, if a port rate shaper is configured, that value overrides the rate shaper value of a priority queue if the priority queue's rate shaper is greater than the rate shaper for the port.
- Configured rate shaper values are rounded up to the nearest multiple of 651 Kbps for 1G ports, and for 10G ports if the configured rate is less than 1.3 Gbps. On 10G ports, if the configured rate is more than 1.3 Gbps, then the values are rounded up to the nearest multiple of 41664 Kbps.

Configuring Outbound Rate Shaping for a Port

To configure the maximum rate at which outbound traffic is sent out on a port, enter the following:

```
FastIron SuperX Router(config)# interface e 1/2
FastIron SuperX Router(config-if-e1000-2)# rate-limit output shaping 1300
```

The configured 1300 Kbps outbound rate shaping on Port 2 is rounded up to the nearest multiple of 651 Kbps, which is 1302 Kbps. This value is the actual limit on the port for outbound traffic.

Syntax: [no] rate-limit output shaping <value>

You can configure up to 2665845 Kbps for <value>.

Configuring Outbound Rate Shaping for a Specific Priority

To configure the maximum rate at which outbound traffic is sent out on a port's priority queue, enter the following:

```
FastIron SuperX Router(config)# interface e 1/2
FastIron SuperX Router(config-if-e1000-1)# rate-limit output shaping 500 priority 7
```

The configured 500 Kbps limit for outbound traffic on Priority queue 7 on Port 2 is rounded up to the nearest multiple of 651 Kbps, which is 651 Kbps.

Syntax: [no] rate-limit output shaping <value> priority <priority-queue>

You can configure up to 2665845 Kbps for <value>.

Specify 0-7 for <priority-queue>

Configuring Outbound Rate Shaping for a Trunk Port

To configure the maximum rate at which outbound traffic is sent out on a trunk port, enter the following on each trunk port where outbound traffic will be shaped.

```
FastIron SuperX Router(config)# trunk e 1/13 to 1/16
FastIron SuperX Router(config-trunk-13-16)# config-trunk-ind
FastIron SuperX Router(config-trunk-13-16)# rate-limit output shaping ethe 1/15 651
FastIron SuperX Router(config-trunk-13-16)# rate-limit output shaping ethe 1/14 1300
```

An outbound rate shaper is configured on Port 1/14 and Port 1/15. The configured outbound rate shaper (651 Kbps) on Port 1/15 is the maximum rate of outbound traffic that is sent out on that port, since 651 Kbps is a multiple of 651 Kbps.

The configured 1300 Kbps limit on Port 14 is rounded up to 1302 Kbps.

Syntax: [no] rate-limit output shaping ethernet [<slotnum>/]<portnum> <value>

The <slotnum> parameter is required on chassis devices.

You can configure up to 2665845 Kbps for <value>.

Displaying Rate Shaping Configurations

To display the configured outbound rate shaper on a device, enter the following command:

```
FESX424 Router#show rate-limit output-shaping
Outbound Rate Shaping Limits in Kbps:
 Port   PortMax   Prio0   Prio1   Prio2   Prio3   Prio4   Prio5   Prio6   Prio7
    1         -       -       -       -       -       -       -       -     651
    2      1302       -       -       -       -       -       -       -       -
   15       651       -       -       -       -       -       -       -       -
```

Syntax: show rate-limit output-shaping

The display lists the ports on a device, the configured outbound rate shaper on a port and for a priority for a port.

VSRP, VRRP, and VRRP-E Scale Timer

The **scale-timer** command allows you to adjust the timers for the Hello interval, Dead interval, Backup Hello interval, and Hold-down interval for VSRP, VRRP, and VRRP-E. Each timer's value is divided by the scale timer value. Increasing or decreasing the scale timer adjusts the value of the individual timers so you can easily change all the timers while preserving the ratios among their values.

To change the scale timer, enter a command such as the following at the global CONFIG level of the CLI:

```
FESX424 Router(config)# scale-timer 2
```

This command changes the scale timer to 2. All VSRP, VRRP, and VRRP-E timer values will be divided by 2.

Syntax: [no] scale-timer <num>

You can specify 1 – 10 for <num>. The default is 1.

In addition to this change, the **show ip vrrp-ext** and **show ip vrrp** have been enhanced to display the configured Hello interval, Dead interval, Backup Hello interval, and Hold-down interval time values in milliseconds instead of seconds. Also the time displayed for the "next hello sent in xxx.xxx.xxx.xxx", "master router xxx.xxx.xxx.xxx expires in", and "backup router xxx.xxx.xxx.xxx expires in" fields is in the hour, minute, seconds, and hundredths second format. For example,

```
FESX424 Router#show ip vrrp-e
Total number of VRRP-Extended routers defined: 1
Interface ethernet v2
 auth-type no authentication
 VRID 2
  state master
  administrative-status enabled
  priority 160
  current priority 160
  hello-interval 3000 msec
  dead-interval 10000 msec
  current dead-interval 10000 msec
  preempt-mode true
  virtual ip address 2.2.2.4
  virtual mac address 02e0.524c.9602
  advertise backup: disabled
  next hello sent in 00:00:03.0
```

VRRP-E Slow Start Timer

In a VRRP-E configuration, if a Master router goes down, the Backup router with the highest priority takes over. When the Master comes back up again, it takes over from the Backup. By default, this transition from Backup back to Master takes place immediately. However, you can configure the VRRP-E slow start timer feature, which causes a specified amount of time to elapse between the time the Master is restored and when it takes over from the Backup. This interval allows time for OSPF convergence when the Master is restored.

To set the VRRP-E slow start timer to 30 seconds, enter the following commands:

```
FESX424 Router(config)# router vrrp-e
FESX424 Router(config-vrrpe-router)# slow-start 30
```

Syntax: [no] slow-start <seconds>

For <seconds>, enter a value from 1 – 255.

When the VRRP-E slow start timer is enabled, if the Master goes down, the Backup takes over immediately. If the Master subsequently comes back up again, the amount of time specified by the VRRP-E slow start timer elapses (in this example, 30 seconds) before the Master takes over from the Backup.

The VRRP-E slow start timer is effective only if another VRRP-E Master (Standby) is detected. It is not effective during the initial boot up.

NOTE: The VRRP-E slow start timer applies only to VRRP-E configurations. It does not apply to VRRP configurations.

Clearing OSPF Information from a Foundry Device

The following kinds of OSPF information can be cleared from a Foundry device’s OSPF link state database and OSPF routing table:

- Routes received from OSPF neighbors. You can clear routes from all OSPF neighbors, or an individual OSPF neighbor, specified either by the neighbor’s IP address or its router ID
- OSPF topology information, including all routes in the OSPF routing table
- All routes in the OSPF routing table that were redistributed from other protocols
- OSPF area information, including routes received from OSPF neighbors within an area, as well as routes imported into the area. You can clear area information for all OSPF areas, or for a specified OSPF area

The OSPF information is cleared dynamically when you enter the command; you do not need to remove statements from the Foundry device’s configuration or reload the software for the change to take effect.

Clearing OSPF Neighbor Information

To clear information on the Foundry device about all OSPF neighbors, enter the following command:

```
FESX424 Router# clear ip ospf neighbor
```

Syntax: clear ip ospf neighbor [ip <ip-addr> | id <ip-addr>] |

This command clears all OSPF neighbors and the OSPF routes exchanged with the neighbors in the Foundry device’s OSPF link state database. After this information is cleared, adjacencies with all neighbors are re-established, and routes with these neighbors exchanged again.

To clear information on the Foundry device about OSPF neighbor 10.10.10.1, enter the following command:

```
FESX424 Router# clear ip ospf neighbor ip 10.10.10.1
```

This command clears the OSPF neighbor and the OSPF routes exchanged with neighbor 10.10.10.1 in the Foundry device's OSPF link state database. After this information is cleared, the adjacency with the neighbor is re-established, and routes are exchanged again.

The neighbor router can be specified either by its IP address or its router ID. To specify the neighbor router using its IP address, use the **ip** <ip-addr> parameter. To specify the neighbor router using its router ID, use the **id** <ip-addr> parameter.

Clearing OSPF Topology Information

To clear OSPF topology information on the Foundry device, enter the following command:

```
FESX424 Router# clear ip ospf topology
```

Syntax: clear ip ospf topology

This command clears all OSPF routes from the OSPF routing table, including intra-area, (which includes ABR and ASBR intra-area routes), inter-area, external type 1, external type 2, OSPF default, and OSPF summary routes.

After you enter this command, the OSPF routing table is rebuilt, and valid routes are recomputed from the OSPF link state database. When the OSPF routing table is cleared, OSPF routes in the global routing table are also recalculated. If redistribution is enabled, the routes are imported again.

Clearing Redistributed Routes from the OSPF Routing Table

To clear all routes in the OSPF routing table that were redistributed from other protocols, enter the following command:

```
FESX424 Router# clear ospf redistribution
```

Syntax: clear ospf redistribution

This command clears all routes in the OSPF routing table that are redistributed from other protocols, including direct connected, static, RIP, BGP, and IS-IS. To import redistributed routes from other protocols, use the **redistribution** command at the OSPF configuration level.

Clearing Information for OSPF Areas

To clear information on the Foundry device about all OSPF areas, enter the following command:

```
FESX424 Router# clear ip ospf
```

This command clears all OSPF areas, all OSPF neighbors, and the entire OSPF routing table. After this information has been cleared, adjacencies with all neighbors are re-established, and all OSPF routes are re-learned.

To clear information on the Foundry device about OSPF area 1, enter the following command:

```
FESX424 Router# clear ip ospf area 1
```

This command clears information about the specified area ID. Information about other OSPF areas is not affected. The command clears information about all OSPF neighbors belonging to the specified area, as well as all routes imported into the specified area. Adjacencies with neighbors belonging to the area are re-established, and routes imported into the area are re-learned.

Syntax: clear ip ospf [area <area-id>]

The <area-id> can be specified in decimal format or in IP address format.

Restricting Telnet and SSH Access Based on a Client's MAC Address

You can restrict remote management access to the Foundry device based on the MAC address of a connecting client. This enhancement applies to Telnet and SSH access to the device. For example, the following command allows Telnet access to the Foundry device only to the host with IP address 209.157.22.39 ***and*** MAC address 0007.e90f.e9a0:

```
FESX424 Router(config)# telnet-client 209.157.22.39 0007.e90f.e9a0
```

Syntax: [no] telnet-client <ip-addr> <mac-addr>

The following command allows Telnet access to the Foundry device to a host with any IP address and MAC address 0007.e90f.e9a0:

```
FESX424 Router(config)# telnet-client any 0007.e90f.e9a0
```

Syntax: [no] telnet-client any <mac-addr>

To allow SSH access to the Foundry device only to the host with IP address 209.157.22.39, enter the following command:

```
FESX424 Router(config)# ip ssh client 209.157.22.39
```

Syntax: [no] ip ssh client <ip-addr>

To allow SSH access to the Foundry device only to the host with IP address 209.157.22.39 ***and*** MAC address 0007.e90f.e9a0, enter the following command:

```
FESX424 Router(config)# ip ssh client 209.157.22.39 0007.e90f.e9a0
```

Syntax: [no] ip ssh client <ip-addr> <mac-addr>

To allow SSH access to the Foundry device to a host with any IP address and MAC address 0007.e90f.e9a0, enter the following command:

```
FESX424 Router(config)# ip ssh client any 0007.e90f.e9a0
```

Syntax: [no] ip ssh client any <mac-addr>

Specifying a Minimum Number of Ports for a Trunk Group

You can configure the Foundry device to disable all of the ports in a trunk group when the number of active member ports drops below a specified threshold value. For example, if a trunk group has 4 ports, and the threshold for the trunk group is 3, then the trunk group is disabled if the number of available ports in the trunk group drops below 3. If the trunk group is disabled, then traffic is forwarded over a different link or trunk group.

For example, the following commands establish a trunk group consisting of 4 ports, then establish a threshold for this trunk group of 3 ports.

```
FastIron SuperX Router(config)# trunk e 3/31 to 3/34
FastIron SuperX Router(config-trunk-3/31-3/34)# threshold 3
```

In this example, if the number of active ports drops below 3, then all the ports in the trunk group are disabled.

Syntax: [no] threshold <number>

You can specify a threshold from 1 (the default) up to the number of ports in the trunk group.

Notes:

- The **disable module** command can be used to disable the ports on a module. However, on 10 Gigabit modules, the **disable module** command does not cause the remote connection to be dropped. If a trunk group consists of 10 Gigabit ports, and you use the **disable module** command to disable ports in the trunk group, which then causes the number of active ports in the trunk group to drop below the threshold value, the trunk group is not disabled.
- If you establish a threshold for a trunk used in conjunction with the Metro Ring Protocol (MRP) on 10 Gigabit interfaces, then you must also enable Link Fault Signalling (LFS).
- If you specify a threshold for a trunk group, the other end of the trunk group must also have the same threshold configuration.

Protected Link Groups

A protected link group minimizes disruption to the network by protecting critical links from loss of data and power. In a protected link group, one port in the group acts as the primary or active link, and the other ports act as secondary or standby links. The active link carries the traffic. If the active link goes down, one of the standby links takes over.

During normal operation, the active port in a protected link group is enabled and the standby ports are logically disabled. If the active port fails, the Foundry device immediately enables one of the standby ports, and switches traffic to the standby port. The standby port becomes the new, active port.

About Active Ports

When you create a protected link group, you can optionally specify which port in the protected link group is the active port. If you do not explicitly configure an active port, the Foundry device dynamically assigns one. A dynamic active port is the first port in the protected link group that comes up (usually the lowest numbered port in the group).

Static and dynamic active ports operate as follows:

- A static active port (an active port that you explicitly configured) preempts other ports in the protected link group. So, if a static active link comes back up after a failure, the Foundry device will revert to this link as the active link.
- A dynamic active port (an active port assigned by the software) is non-preemptive. Therefore, if a dynamic active link comes back up after a failure, the Foundry device does not revert to this link, but continues carrying traffic on the current active link.

Using UDLD with Protected Link Groups

You can use Uni-directional Link Detection (UDLD) with protected link groups to detect uni-directional link failures and to improve the speed at which the device detects a failure in the link. Use UDLD and protected link groups simultaneously when the FastIron X-Series device is connected to a device with slower link detection times.

Configuration Notes

- You can configure a maximum of 32 protected link groups.
- There is no restriction on the number of ports in a protected link group.
- Each port can belong to one protected link group at a time.
- There is no restriction on the type of ports in a protected link group. A protected link group can consist of Gigabit fiber ports, 10/100/1000 copper ports, and 10/100 ports, or any combination thereof.
- There is no restriction on the properties of ports in a protected link group. For example, member ports can be in the same VLAN or in different VLANs.

Creating a Protected Link Group and Assigning an Active Port

To create a protected link group:

1. Specify the member ports in the protected link group. Enter a command such as the following:

   ```
   FESX424 Switch(config)# protected-link-group 10 e 1 to 4
   ```

2. Optionally specify which port will be the active port for the protected link group. Enter a command such as the following:

   ```
   FESX424 Switch(config)# protected-link-group 10 active-port e 1
   ```

   NOTE: If you do not explicitly configure an active port, the Foundry device automatically assigns one as the first port in the protected link group to come up.

These commands configure port e1 as the active port and ports e2 – e4 as standby ports. If port 1 goes down, the Foundry device enables the first available standby port, and switches the traffic to that port. Since the above configuration consists of a statically configured active port, the active port preempts other ports in the protected link group. See “About Active Ports” on page 36.

Syntax: [no] protected-link-group <group-ID> ethernet [<slotnum>/]<portnum> to [<slotnum>/]<portnum>

The <group-ID> parameter specifies the protected link group number. Enter a number from 1 – 32.

Each **ethernet** parameter introduces a port group.

The [<slotnum>/]portnum> **to** [<slotnum>/]<portnum> parameters specify the ports in the protected link group. The <slotnum> parameter is required on chassis devices.

Syntax: [no] protected-link-group <group-ID> active-port ethernet <[slotnum/]portnum>

The <group-ID> parameter specifies the protected link group number. Enter a number from 1 – 32.

The **active-port ethernet** [<slotnum>/]portnum> parameter defines the active port. The <slotnum> parameter is required on chassis devices.

Viewing Information about Protected Link Groups

You can view protected link group information using the **show protected-link-group** command. The following shows an example output.

```
FESX424 Switch# show protected-link-group

Group ID: 1
Member Port(s): ethe 1 to 7
Configured Active Port: 7
Current Active Port: 7
Standby Port(s): ethe 5

Total Number of Protected Link Groups: 1
```

Syntax: show protected-link-group [group-ID]

Table 8: CLI Display of Protected Link Group Information

This Field...	Displays...
Group ID	The ID number of the protected link group.
Member Port(s)	The ports that are members of the protected link group.
Configured Active Port	The statically configured active port. If you do not statically configure an active port, this value will be "None".
Current Active Port	The current active port for the protected link group. If all member ports are down, this value will be "None".
Standby Port(s)	The member ports that are on standby.

ACL Logging

You may want the software to log entries in the Syslog for packets that are denied by ACL filters. ACL logging is disabled by default; it must be explicitly enabled on a port.

When you enable logging for ACL entries, statistics for packets that match the deny conditions of the ACL entries are logged. For example, if you configure a standard ACL entry to deny all packets from source address 209.157.22.26, statistics for packets that are explicitly denied by the ACL entry are logged in the Syslog buffer and in SNMP traps sent by the Foundry device.

The first time an ACL entry denies a packet, the software immediately generates a Syslog entry and SNMP trap. The software also starts a five-minute timer. The timer keeps track of all packets explicitly denied by the ACL entries. After five minutes, the software generates a single Syslog entry for each ACL entry that has denied a packet. The message indicates the number of packets denied by the ACL entry during the previous five minutes.

If no ACL entries explicitly deny packets during an entire five-minute timer interval, the timer stops. The timer restarts when an ACL entry explicitly denies a packet.

NOTE: The timer for logging packets denied by Layer 2 filters is separate.

Configuration Notes

Note the following before configuring ACL logging:

- You can enable ACL logging on physical and virtual interfaces.
- ACL logging logs denied packets only.
- When ACL logging is disabled, packets that match the ACL rule are forwarded or dropped in hardware. When ACL logging is enabled, all packets that match the ACL deny rule are sent to the CPU. When ACL logging is enabled, Foundry recommends that you configure a traffic conditioner then link the ACL to the traffic conditioner to prevent CPU overload. For example:

  ```
  FastIron SuperX Switch(config)# traffic-policy TPD1 rate-limit fixed 100 exceed-
  action drop
  FastIron SuperX Switch(config)# access-list 101 deny ip host 210.10.12.2 any
  traffic-policy TPD1 log
  ```

- ACL logging is intended for debugging purpose. Foundry recommends that you disable ACL logging after the debug session is over.

Enabling ACL Logging

To enable ACL logging, complete the following steps:

- Create ACL entries with the log option
- Enable ACL logging on individual ports
- Bind the ACLs to the ports on which ACL logging is enabled

The following shows an example configuration.

```
FastIron SuperX Switch(config)# access-list 1 deny host 209.157.22.26 log
FastIron SuperX Switch(config)# access-list 1 deny 209.157.29.12 log
FastIron SuperX Switch(config)# access-list 1 deny host IPHost1 log
FastIron SuperX Switch(config)# access-list 1 permit any
FastIron SuperX Switch(config)# interface e 1/4
FastIron SuperX Switch(config-if-e1000-4)# acl-logging
FastIron SuperX Switch(config-if-e1000-4)# ip access-group 1 in
```

The above commands create ACL entries that include the log option, enable ACL logging on interface e 1/4, then bind the ACL to interface e 1/4. Statistics for packets that match the deny statements will be logged.

NOTE: If the same ACL is used on multiple ports, you must enable or disable ACL logging on the lowest-numbered port. The following shows an example configuration:

```
FastIron SuperX Switch(config)# access-list 1 deny host 209.157.22.26 log
FastIron SuperX Switch(config)# interface e 2/4
FastIron SuperX Switch(config-if-e2000-4)# ip access-group 1 in
FastIron SuperX Switch(config)# interface e 1/4
FastIron SuperX Switch(config-if-e1000-4)# ip access-group 1 in
FastIron SuperX Switch(config)# interface e 1/6
FastIron SuperX Switch(config-if-e1000-6)# ip access-group 1 in
FastIron SuperX Switch(config)# interface e 1/4
FastIron SuperX Switch(config-if-e1000-4)# acl-logging
FastIron SuperX Switch(config-if-e1000-4)# no acl-logging
```

Displaying ACL Log Entries

The first time an entry in an ACL permits or denies a packet and logging is enabled for that entry, the software generates a Syslog message and an SNMP trap. Messages for packets permitted or denied by ACLs are at the warning level of the Syslog.

When the first Syslog entry for a packet permitted or denied by an ACL is generated, the software starts an ACL timer. After this, the software sends Syslog messages every one to ten minutes, depending on the value of the timer interval. If an ACL entry does not permit or deny any packets during the timer interval, the software does not generate a Syslog entry for that ACL entry.

NOTE: For an ACL entry to be eligible to generate a Syslog entry for denied packets, logging must be enabled for the entry. The Syslog contains entries only for the ACL entries that deny packets and have logging enabled.

To display Syslog entries, use one of the following methods.

Enter the following command from any CLI prompt:

```
FESX424 Router(config-if-e1000-4)#sh log
Syslog logging: enabled (0 messages dropped, 2 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 9 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning

Dynamic Log Buffer (50 lines):
0d00h12m18s:W:ACL: ACL: List 122 denied tcp 20.20.15.6(0)(Ethernet 4
0000.0804.0101) -> 20.20.18.6(0), 1 event(s)
0d00h12m18s:W:ACL: ACL: List 122 denied tcp 20.20.15.2(0)(Ethernet 4
0000.0804.0101) -> 20.20.18.2(0), 1 event(s)
0d00h12m18s:W:ACL: ACL: List 122 denied tcp 20.20.15.4(0)(Ethernet 4
0000.0804.0101) -> 20.20.18.4(0), 1 event(s)
0d00h12m18s:W:ACL: ACL: List 122 denied tcp 20.20.15.3(0)(Ethernet 4
0000.0804.0101) -> 20.20.18.3(0), 1 event(s)
0d00h12m18s:W:ACL: ACL: List 122 denied tcp 20.20.15.5(0)(Ethernet 4
0000.0804.0101) -> 20.20.18.5(0), 1 event(s)
0d00h12m18s:I:ACL: 122 applied to port 4 by  from console session
0d00h10m12s:I:ACL: 122 removed from port 4 by  from console session
0d00h09m56s:I:ACL: 122 removed from port 4 by  from console session
0d00h09m38s:I:ACL: 122 removed from port 4 by  from console session
```

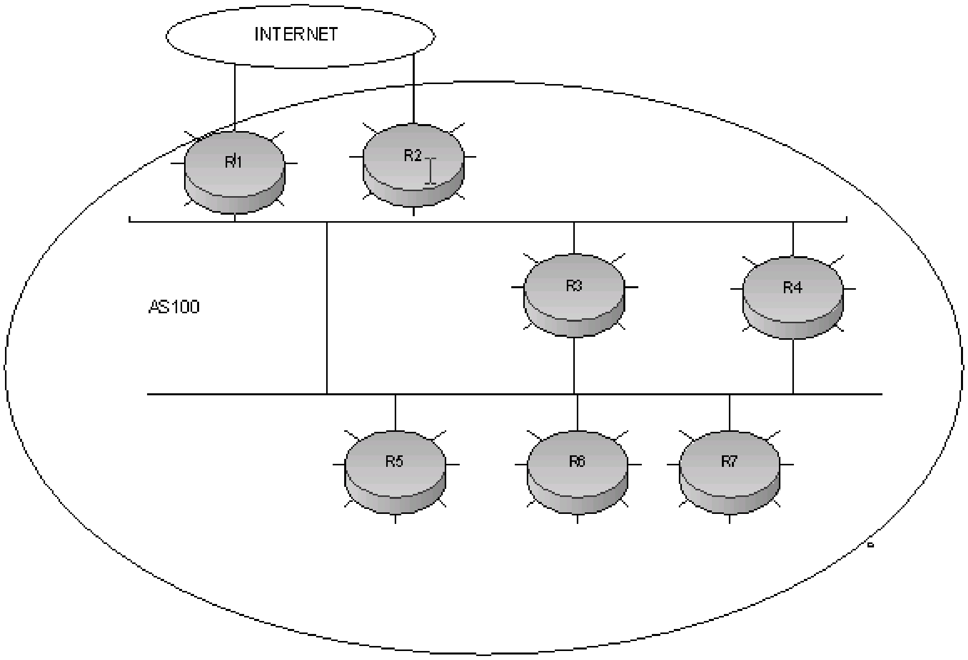
BGP Null0 Routing

The null0 routes used to be treated as invalid routes for BGP next hop resolution. Now BGP can use the null0 route to resolve its next hop. Thus, null0 route in the routing table (for example, static route) is considered as a valid route by BGP. If the next hop for BGP resolves into a null0 route, the BGP route is also installed as a null0 route in the routing table.

The null0 routing feature allows network administrators to block certain network prefixes, by using null0 routes and route-maps. The combined use of null0 routes and route maps blocks traffic from a particular network prefix, telling a remote router to drop all traffic for this network prefix by redistributing a null0 route into BGP.

Figure 8 shows a topology for a null0 routing application example.

Figure 8 Example Null0 Routing Application



The following steps configure a null0 routing application for stopping denial of service attacks from remote host(s) on the internet.

Configuration Steps

1. Select one router, Router 6, to distribute null0 routes throughout the BGP network.
2. Configure a route-map to match a particular tag (50) and set the next-hop address to an unused network address (199.199.1.1).
3. Set the local-preference to a value higher than any possible internal/external local-preference (50).
4. Complete the route map by setting origin to IGP.
5. On Router 6, redistribute the static routes into BGP, using route-map <route-map-name> (redistribute static route-map block user).
6. On Router 1, the router facing the internet, configure a null0 route matching the next-hop address in the route-map (ip route 199.199.1.1/32 null0).
7. Repeat step 3 for all routers interfacing with the internet (edge corporate routers). In this case, Router 2 has the same null0 route as Router 1.
8. On Router 6, configure the network prefixes associated with the traffic you want to drop. The static route IP address references a destination address. You are required to point the static route to the egress port, for example, Ethernet 3/7, and specify the tag 50, matching the route-map configuration.

Configuration Examples

Router 6

The following configuration defines specific prefixes to filter:

```
FastIron SuperX Router(config)# ip route 110.0.0.40/29 ethernet 3/7 tag 50
FastIron SuperX Router(config)# ip route 115.0.0.192/27 ethernet 3/7 tag 50
FastIron SuperX Router(config)# ip route 120.014.0/23 ethernet 3/7 tag 50
```

The following configuration redistributes routes into BGP:

```
FastIron SuperX Router(config)# router bgp
FastIron SuperX Router(config-bgp-router)# local-as 100
FastIron SuperX Router(config-bgp-router)# neighbor <router1_int_ip address> remote-
as 100
FastIron SuperX Router(config-bgp-router)# neighbor <router2_int_ip address> remote-
as 100
FastIron SuperX Router(config-bgp-router)# neighbor <router3_int_ip address> remote-
as 100
FastIron SuperX Router(config-bgp-router)# neighbor <router4_int_ip address> remote-
as 100
FastIron SuperX Router(config-bgp-router)# neighbor <router5_int_ip address> remote-
as 100
FastIron SuperX Router(config-bgp-router)# neighbor <router7_int_ip address> remote-
as 100
FastIron SuperX Router(config-bgp-router)# redistribute static route-map blockuser
FastIron SuperX Router(config-bgp-router)# exit
```

The following configuration defines the specific next hop address and sets the local preference to preferred:

```
FastIron SuperX Router(config)# route-map blockuser permit 10
FastIron SuperX Router(config-routemap blockuser)# match tag 50
FastIron SuperX Router(config-routemap blockuser)# set ip next-hop 199.199.1.1
FastIron SuperX Router(config-routemap blockuser)# set local-preference 1000000
FastIron SuperX Router(config-routemap blockuser)# set origin igp
FastIron SuperX Router(config-routemap blockuser)# exit
```

Router 1

The following configuration defines the null0 route to the specific next hop address. The next hop address 199.199.1.1 points to the null0 route:

```
FastIron SuperX Router(config)# ip route 199.199.1.1/32 null0
FastIron SuperX Router(config)# router bgp
FastIron SuperX Router(config-bgp-router)# local-as 100
FastIron SuperX Router(config-bgp-router)# neighbor <router2_int_ip address> remote-
as 100
FastIron SuperX Router(config-bgp-router)# neighbor <router3_int_ip address> remote-
as 100
FastIron SuperX Router(config-bgp-router)# neighbor <router4_int_ip address> remote-
as 100
FastIron SuperX Router(config-bgp-router)# neighbor <router5_int_ip address> remote-
as 100
FastIron SuperX Router(config-bgp-router)# neighbor <router6_int_ip address> remote-
as 100
FastIron SuperX Router(config-bgp-router)# neighbor <router7_int_ip address> remote-
as 100
```

Router 2

The following configuration defines a null0 route to the specific next hop address. The next hop address 199.199.1.1 points to the null0 route, which gets blocked:

```
FastIron SuperX Router(config)# ip route 199.199.1.1/32 null0
FastIron SuperX Router(config)# router bgp
FastIron SuperX Router(config-bgp-router)# local-as 100
FastIron SuperX Router(config-bgp-router)# neighbor <router1_int_ip address> remote-
as 100
FastIron SuperX Router(config-bgp-router)# neighbor <router3_int_ip address> remote-
as 100
FastIron SuperX Router(config-bgp-router)# neighbor <router4_int_ip address> remote-
as 100
```

```
FastIron SuperX Router (config-bgp-router)# neighbor <router5_int_ip address>
remote-as 100
FastIron SuperX Router(config-bgp-router)# neighbor <router6_int_ip address> remote-
as 100
FastIron SuperX Router(config-bgp-router)# neighbor <router7_int_ip address> remote-
as 100
```

Show Commands

After configuring the null0 application, you can display the output:

Router 6

Show ip route static output for Router 6:

```
FastIron SuperX Switch#show ip route static

Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost - Dist/Metric
          Destination          Gateway         Port         Cost       Type
1       110.0.0.40/29          DIRECT          eth 3/7      1/1        S
2       115.0.0.192/27         DIRECT          eth 3/7      1/1        S
3       120.0.14.0/23          DIRECT          eth 3/7      1/1        S
```

Router 1 and 2

Show ip route static output for Router 1 and Router 2:

```
FastIron SuperX Router# show ip route static

 Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost - Dist/Metric
             Destination        Gateway         Port        Cost       Type
1            199.199.1.1/32     DIRECT          drop        1/1        S
```

Router 6

Show BGP routing table output for Router-6

```
FastIron SuperX Router-6#show ip bgp route

Total number of BGP Routes: 126
Status A:AGGREGATE B:BEST b:NOT-INSTALLED-BEST C:CONFED_EBGP D:DAMPED  E:EBGP
H:HISTORY I:IBGP L:LOCAL M:MULTIPATH S:SUPPRESSED s:STALE
       Prefix             Next Hop        Metric     LocPrf     Weight Status
1      30.0.1.0/24        40.0.1.3        0          100        0      BI
         AS_PATH:
.             ..                       .                       .           .     .
9      110.0.0.16/30      90.0.1.3                   100        0      I
         AS_PATH: 85
10     110.0.0.40/29      199.199.1.1/32  1        1000000 32768  BL
         AS_PATH:
11     110.0.0.80/28      90.0.1.3                   100        0      I
 .           ..                       .                       .          .       .
 .           ..                       .                      .          .       .
36     115.0.0.96/28      30.0.1.3                   100        0      I
         AS_PATH: 50
37     115.0.0.192/27     199.199.1.1/32  1       10000000 32768  BL
         AS_PATH:
 .             ..                    .                          .          .     .
64     120.0.7.0/24       70.0.1.3                   100        0      I
         AS_PATH: 10
65     120.0.14.0/23      199.199.1.1/32  1        1000000 32768  BL
         AS_PATH: ..                  .                       .            .     .
```

Router 1 and 2

The **show ip route** output for Router 1 and Router 2 shows "drop" under the Port column for the network prefixes you configured with null0 routing:

```
FastIron SuperX Router# show ip route

Total number of IP routes: 133
 Type Codes - B:BGP  D:Connected  S:Static  R:RIP  O:OSPF;   Cost - Dist/Metric
        Destination        Gateway         Port        Cost       Type
1       9.0.1.24/32        DIRECT          loopback 1  0/0     D
2       30.0.1.0/24        DIRECT          eth 2/7     0/0        D
3       40.0.1.0/24        DIRECT          eth 2/1     0/0        D
.
13      110.0.0.6/31       90.0.1.3        eth 2/2     20/1       B
14      110.0.0.16/30      90.0.1.3        eth 2/2     20/1       B
15      110.0.0.40/29      DIRECT          drop        200/0      B
.          ..               .                .          .         .
42      115.0.0.192/27     DIRECT          drop        200/0      B
43      115.0.1.128/26     30.0.1.3        eth 2/7     20/1       B
.          ..               .               .           .         .
69      120.0.7.0/24       70.0.1.3        eth 2/10    20/1       B
70      120.0.14.0/23      DIRECT          drop        200/0      B
.          ..               .                .          .         .
.          ..               .                .          .         .
131     130.144.0.0/12     80.0.1.3        eth 3/4     20/1       B
132     199.199.1.1/32     DIRECT          drop        1/1        S
```

PBR

Policy-Based Routing (PBR) allows you to use ACLs and route maps to selectively modify and route IP packets in hardware. The ACLs classify the traffic. Route maps that match on the ACLs set routing attributes for the traffic.

A PBR policy specifies the next hop for traffic that matches the policy. Using standard ACLs with PBR, you can route IP packets based on their source IP address. With extended ACLs, you can route IP packets based on all of the clauses in the extended ACL.

You can configure the Foundry device to perform the following types of PBR based on a packet's Layer 3 and Layer 4 information:

- Select the next-hop gateway.
- Send the packet to the null interface (null0).

When a PBR policy has multiple next hops to a destination, PBR selects the first live next hop specified in the policy that is up. If none of the policy's direct routes or next hops are available, the packet is routed in the normal way.

Configuration Considerations

- A PBR policy on an interface takes precedence over a global PBR policy.
- You cannot apply PBR on a port if that port already has ACLs, ACL-based rate limiting, DSCP-based QoS, MAC filtering, or IP Source Guard.
- The number of route maps that you can define is limited by the system memory. When a route map is used in a PBR policy, the PBR policy uses up to 6 instances of a route map, up to 6 ACLs in a matching policy of each route map instance, and up to 6 next hops in a set policy of each route map instance.
- ACLs with the **log** option configured should not be used for PBR purposes.
- PBR ignores explicit or implicit **deny ip any any** ACL entries, to ensure that for route maps that use multiple

ACLs, the traffic is compared to all the ACLs. PBR also ignores any deny clauses in an ACL. Traffic that matches a deny clause is routed normally using Layer 3 paths.

- PBR always selects the first next hop from the next hop list that is up. If a PBR policy's next hop goes down, the policy uses another next hop if available. If no next hops are available, the device routes the traffic in the normal way.
- PBR is not supported for fragmented packets. If the PBR’s ACL filters on Layer 4 information like TCP/UDP ports, fragmented packed are routed normally.
- You can change route maps or ACL definitions dynamically and do not need to rebind the PBR policy to an interface.

Configuring a PBR Policy

To configure PBR, you define the policies using IP ACLs and route maps, then enable PBR globally or on individual interfaces. The device programs the ACLs into the packet processor on the interfaces and routes traffic that matches the ACLs according to the instructions in the route maps.

To configure a PBR policy:

- Configure ACLs that contain the source IP addresses for the IP traffic you want to route using PBR.
- Configure a route map that matches on the ACLs and sets the route information.
- Apply the route map to an interface.

Configure the ACLs

PBR uses route maps to change the routing attributes in IP traffic. This section shows an example of how to configure a standard ACL to identify the source subnet for IP traffic. See the *Foundry FastIron X-Series Configuration Guide* for details on how to configure ACLs.

To configure a standard ACL to identify a source subnet, enter a command such as the following:

```
FESX424 Router(config)# access-list 99 permit 209.157.23.0 0.0.0.255
```

The command in this example configures a standard ACL that permits traffic from subnet 209.157.23.0/24. After you configure a route map that matches based on this ACL, the software uses the route map to set route attributes for the traffic, thus enforcing PBR.

NOTE: Do not use an access group to apply the ACL to an interface. Instead, use a route map to apply the ACL globally or to individual interfaces for PBR, as shown in the following sections.

Syntax: [no] access-list <num> deny | permit <source-ip> | <hostname> <wildcard>

or

Syntax: [no] access-list <num> deny | permit <source-ip>/<mask-bits> | <hostname>

Syntax: [no] access-list <num> deny | permit host <source-ip> | <hostname>

Syntax: [no] access-list <num> deny | permit any

The <num> parameter is the access list number and can be from 1 – 99.

The **deny | permit** parameter indicates whether packets that match a policy in the access list are denied (dropped) or permitted (forwarded).

NOTE: If you are configuring the ACL for use in a route map, always specify **permit**. Otherwise, the Foundry device will ignore deny clauses and packets that match deny clauses are routed normally.

The <source-ip> parameter specifies the source IP address. Alternatively, you can specify the host name.

NOTE: To specify the host name instead of the IP address, the host name must be configured using the Foundry device's DNS resolver. To configure the DNS resolver name, use the **ip dns server-address…** command at the global CONFIG level of the CLI.

The <wildcard> parameter specifies the mask value to compare against the host address specified by the <source-ip> parameter. The <wildcard> is a four-part value in dotted-decimal notation (IP address format) consisting of ones and zeros. Zeros in the mask mean the packet's source address must match the <source-ip>. Ones mean any value matches. For example, the <source-ip> and <wildcard> values 209.157.22.26 0.0.0.255 mean that all hosts in the Class C subnet 209.157.22.x match the policy.

If you prefer to specify the wildcard (mask value) in CIDR format, you can enter a forward slash after the IP address, then enter the number of significant bits in the mask. For example, you can enter the CIDR equivalent of "209.157.22.26 0.0.0.255" as "209.157.22.26/24". The CLI automatically converts the CIDR number into the appropriate ACL mask (where zeros instead of ones are the significant bits) and changes the non-significant portion of the IP address into zeros. For example, if you specify 209.157.22.26/24 or 209.157.22.26 0.0.0.255, then save the changes to the startup-config file, the value appears as 209.157.22.0/24 (if you have enabled display of subnet lengths) or 209.157.22.0 0.0.0.255 in the startup-config file.

If you enable the software to display IP subnet masks in CIDR format, the mask is saved in the file in "/<mask-bits>" format. To enable the software to display the CIDR masks, enter the **ip show-subnet-length** command at the global CONFIG level of the CLI. You can use the CIDR format to configure the ACL entry regardless of whether the software is configured to display the masks in CIDR format.

NOTE: If you use the CIDR format, the ACL entries appear in this format in the running-config and startup-config files, but are shown with subnet mask in the display produced by the **show ip access-list** command.

The **host** <source-ip> | <hostname> parameter lets you specify a host IP address or name. When you use this parameter, you do not need to specify the mask. A mask of all zeros (0.0.0.0) is implied.

The **any** parameter configures the policy to match on all host addresses.

NOTE: Do not use the **log** option in ACLs that will be used for PBR.

Configure the Route Map

After you configure the ACLs, you can configure a PBR route map that matches based on the ACLs and sets routing information in the IP traffic.

NOTE: The match and set statements described in this section are the only route-map statements supported for PBR. Other route-map statements described in the documentation apply only to the protocols with which they are described.

To configure a PBR route map, enter commands such as the following:

```
FESX424 Router(config)# route-map test-route permit 99
FESX424 Router(config-routemap test-route)# match ip address 99
FESX424 Router(config-routemap test-route)# set ip next-hop 192.168.2.1
FESX424 Router(config-routemap test-route)# exit
```

The commands in this example configure an entry in a route map named "test-route". The **match** statement matches on IP information in ACL 99. The **set** statement changes the next-hop IP address for packets that match to 192.168.2.1.

Syntax: [no] route-map <map-name> permit | deny <num>

The <map-name> is a string of characters that names the map. Map names can be up to 32 characters in length. You can define an unlimited number of route maps on the Foundry device, as long as system memory is available.

The **permit | deny** parameter specifies the action the Foundry device will take if a route matches a match statement.

- If you specify **deny**, the Foundry device does not apply a PBR policy to packets that match the ACLs in a match clause. Those packets are routed normally,

- If you specify **permit**, the Foundry device applies the match and set statements associated with this route map instance.

The <num> parameter specifies the instance of the route map you are defining. Routes are compared to the instances in ascending numerical order. For example, a route is compared to instance 1, then instance 2, and so on.

PBR uses up to six route map instances for comparison and ignores the rest.

Syntax: [no] match ip address <ACL-num-or-name>

The <ACL-num> parameter specifies a standard or extended ACL number or name.

Syntax: [no] set ip next hop <ip-addr>

This command sets the next-hop IP address for traffic that matches a match statement in the route map.

Syntax: [no] set interface null0

This command sends the traffic to the null0 interface, which is the same as dropping the traffic.

Enabling PBR

After you configure the ACLs and route map entries, you can enable PBR globally, on individual interfaces, or both as described in this section. To enable PBR, you apply a route map you have configured for PBR globally or locally.

Enabling PBR Globally

To enable PBR globally, enter a command such as the following at the global CONFIG level:

```
FESX424 Router(config)# ip policy route-map test-route
```

This command applies a route map named “test-route” to all interfaces on the device for PBR.

Syntax: ip policy route-map <map-name>

Enabling PBR Locally

To enable PBR locally, enter commands such as the following:

```
FESX424 Router(config)# interface ve 1
FESX424 Router(config-vif-1)# ip policy route-map test-route
```

The commands in this example change the CLI to the Interface level for virtual interface 1, then apply the “test-route” route map to the interface. You can apply a PBR route map to Ethernet ports or virtual interfaces.

Syntax: ip policy route-map <map-name>

Enter the name of the route map you want to use for the route-map <map-name> parameter.

Configuration Examples

This section presents configuration examples for:

- “Basic Example” on page 48
- “Setting the Next Hop” on page 48
- “Setting the Output Interface to the Null Interface” on page 49
- “Trunk Formation” on page 49

Basic Example

The following commands configure and apply a PBR policy that routes HTTP traffic received on virtual routing interface 1 from the 10.10.10.x/24 network to 5.5.5.x/24 through next-hop IP address 1.1.1.1/24 or, if 1.1.1.x is unavailable, through 2.2.2.1/24.

```
FastIron SuperX Router(config)# access-list 101 permit tcp 10.10.10.0 0.0.0.255 eq
http 5.5.5.0 0.0.0.255
FastIron SuperX Router(config)# route-map net10web permit 101
FastIron SuperX Router(config-routemap net10web)# match ip address 101
FastIron SuperX Router(config-routemap net10web)# set ip next-hop 1.1.1.1
FastIron SuperX Router(config-routemap net10web)# set ip next-hop 2.2.2.2
FastIron SuperX Router(config-routemap net10web)# exit
FastIron SuperX Router(config)# vlan 10
FastIron SuperX Router(config-vlan-10)# tagged ethernet 1/1 to 1/4

FastIron SuperX Router(config-vlan-10)# router-interface ve 1
FastIron SuperX Router(config)# interface ve 1
FastIron SuperX Router(config-vif-1)# ip policy route-map net10web
```

Syntax: `[no] route-map <map-name> permit | deny <num>`

Syntax: [no] set ip next hop <ip-addr>

This command sets the next-hop IP address for traffic that matches a match statement in the route map.

Setting the Next Hop

The following commands configure the Foundry device to apply PBR to traffic from IP subnets 209.157.23.x, 209.157.24.x, and 209.157.25.x. In this example, route maps specify the next-hop gateway for packets from each of these subnets.

- Packets from 209.157.23.x are sent to 192.168.2.1.
- Packets from 209.157.24.x are sent to 192.168.2.2.
- Packets from 209.157.25.x are sent to 192.168.2.3.

The following commands configure three standard ACLs. Each ACL contains one of the ACLs listed above. Make sure you specify **permit** instead of deny in the ACLs, so that the Foundry device permits the traffic that matches the ACLs to be further evaluated by the route map. If you specify **deny**, the traffic that matches the deny statements are routed normally. Notice that these ACLs specify **any** for the destination address.

```
FastIron SuperX Router(config)# access-list 50 permit 209.157.23.0 0.0.0.255
FastIron SuperX Router(config)# access-list 51 permit 209.157.24.0 0.0.0.255
FastIron SuperX Router(config)# access-list 52 permit 209.157.25.0 0.0.0.255
```

The following commands configure three entries in a route map called “test-route”. The first entry (permit 50) matches on the IP address information in ACL 50 above. For IP traffic from subnet 209.157.23.0/24, this route map entry sets the next-hop IP address to 192.168.2.1.

```
FastIron SuperX Router(config)# route-map test-route permit 50
FastIron SuperX Router(config-routemap test-route)# match ip address 50
FastIron SuperX Router(config-routemap test-route)# set ip next-hop 192.168.2.1
FastIron SuperX Router(config-routemap test-route)# exit
```

The following commands configure the second entry in the route map. This entry (permit 51) matches on the IP address information in ACL 51 above. For IP traffic from subnet 209.157.24.0/24, this route map entry sets the next-hop IP address to 192.168.2.2.

```
FastIron SuperX Router(config)# route-map test-route permit 51
FastIron SuperX Router(config-routemap test-route)# match ip address 51
FastIron SuperX Router(config-routemap test-route)# set ip next-hop 192.168.2.2
FastIron SuperX Router(config-routemap test-route)# exit
```

The following commands configure the third entry in the test-route route map. This entry (permit 52) matches on the IP address information in ACL 52 above. For IP traffic from subnet 209.157.25.0/24, this route map entry sets the next-hop IP address to 192.168.2.3.

```
FastIron SuperX Router(config)# route-map test-route permit 52
FastIron SuperX Router(config-routemap test-route)# match ip address 52
FastIron SuperX Router(config-routemap test-route)# set ip next-hop 192.168.2.3
FastIron SuperX Router(config-routemap test-route)# exit
```

The following command enables PBR by globally applying the test-route route map to all interfaces.

```
FastIron SuperX Router(config)# ip policy route-map test-route
```

Alternatively, you can enable PBR on specific interfaces, as shown in the following example. The commands in this example configure IP addresses in the three source subnets identified in ACLs 50, 51, and 52, then apply route map test-route to the interface.

```
FastIron SuperX Router(config)# interface ve 1
FastIron SuperX Router(config-vif-1)# ip address 209.157.23.1/24
FastIron SuperX Router(config-vif-1)# ip address 209.157.24.1/24
FastIron SuperX Router(config-vif-1)# ip address 209.157.25.1/24
FastIron SuperX Router(config-vif-1)# ip policy route-map test-route
```

Setting the Output Interface to the Null Interface

The following commands configure a PBR policy to send all traffic from 192.168.1.204/32 to the null interface, thus dropping the traffic instead of forwarding it.

```
FastIron SuperX Router(config)# access-list 56 permit 209.168.1.204 0.0.0.0
```

The following commands configure an entry in a route map called “file-13”. The first entry (permit 56) matches on the IP address information in ACL 56 above. For IP traffic from the host 209.168.1.204/32, this route map entry sends the traffic to the null interface instead of forwarding it, thus sparing the rest of the network the unwanted traffic.

```
FastIron SuperX Router(config)# route-map file-13 permit 56
FastIron SuperX Router(config-routemap file-13)# match ip address 56
FastIron SuperX Router(config-routemap file-13)# set interface null0
FastIron SuperX Router(config-routemap file-13)# exit
```

The following command enables PBR by globally applying the route map to all interfaces.

```
FastIron SuperX Router(config)# ip policy route-map file-13
```

Alternatively, you can enable PBR on specific interfaces, as shown in the following example. The commands in this example configure IP addresses in the source subnet identified in ACL 56, then apply route map file-13 to the interface.

```
FastIron SuperX Router(config)# interface ethernet 3/11
FastIron SuperX Router(config-if-e10000-3/11)# ip address 192.168.1.204/32
FastIron SuperX Router(config-if-e10000-3/11)# ip policy route-map file-13
```

Trunk Formation

When a trunk is formed, the PBR policy on the primary port applies to all the secondary ports. If a different PBR policy exists on a secondary port at the time of a trunk formation, that policy is overridden by the PBR policy on the primary port. If the primary port does not have a PBR policy, then the secondary ports will not have a PBR policy.

When a trunk is removed, the PBR policy that was applied to the trunk interface is unbound (removed) from former secondary ports. If global PBR is configured, the secondary ports adhere to the global PBR; otherwise, no PBR policy is bound to former secondary ports.

Using ACLs to Filter ARP Packets

You can use ACLs to filter ARP packets. Without this feature, ACLs cannot be used to permit or deny incoming ARP packets. Although an ARP packet contains an IP address just as an IP packet does, an ARP packet is not an IP packet; therefore, it is not subject to normal filtering provided by ACLs.

When a Foundry device receives an ARP request, the source MAC and IP addresses are stored in the device's ARP table. A new record in the ARP table overwrites existing records that contain the same IP address. This behavior can cause a condition called "ARP hijacking", when two hosts with the same IP address try to send an ARP request to the Foundry device.

Normally ARP hijacking is not a problem because IP assignments are done dynamically; however, in some cases, ARP hijacking can occur, such as when a configuration allows a router interface to share the IP address of another router interface. Since multiple VLANs and the router interfaces that are associated with each of the VLANs share the same IP segment, it is possible for two hosts in two different VLANs to fight for the same IP address in that segment. ARP filtering using ACLs protects an IP host's record in the ARP table from being overwritten by a hijacking host. Using ACLs to filter ARP requests checks the source IP address in the received ARP packet. Only packets with the permitted IP address will be allowed to be to be written in the ARP table; others are dropped.

Configuration Considerations:

- This feature is available on all devices running Layer 3 code. This filtering occurs on the management processor.
- The feature is available on physical interfaces and virtual routing interfaces. It is supported on the following physical interface types: Ethernet and trunks.
- ACLs used to filter ARP packets on a virtual routing interface can be inherited from a previous interface if the virtual routing interface is defined as a follower virtual routing interface.

Configuring ACLs for ARP Filtering

To implement the ACL ARP filtering feature, enter commands such as the following:

```
FastIron SuperX Switch(config)# access-list 101 permit ip host 192.168.2.2 any
FastIron SuperX Switch(config)# access-list 102 permit ip host 192.168.2.3 any
FastIron SuperX Switch(config)# access-list 103 permit ip host 192.168.2.4 any
FastIron SuperX Switch(config)# vlan 2
FastIron SuperX Switch(config-vlan-2)# tag ethe 1/1 to 1/2
FastIron SuperX Switch(config-vlan-2)# router-interface ve 2
FastIron SuperX Switch(config-vlan-2)# vlan 3
FastIron SuperX Switch(config-vlan-3)# tag ethe 1/1 to 1/2
FastIron SuperX Switch(config-vlan-3)#router-int ve 3
FastIron SuperX Switch(config-vlan-3)# vlan 4
FastIron SuperX Switch(config-vlan-4)# tag ethe 1/1 to 1/2
FastIron SuperX Switch(config-vlan-4)# router-int ve 4
FastIron SuperX Switch(config-vlan-4)# interface ve 2
FastIron SuperX Switch(config-ve-2)# ip access-group 101 in
FastIron SuperX Switch(config-ve-2)# ip address 192.168.2.1/24
FastIron SuperX Switch(config-ve-2)# ip use-acl-on-arp 103
FastIron SuperX Switch(config-ve-2)# exit
FastIron SuperX Switch(config)# interface ve 3
FastIron SuperX Switch(config-ve-3)# ip access-group 102 in
FastIron SuperX Switch(config-ve-3)# ip follow ve 2
FastIron SuperX Switch(config-ve-3)# ip use-acl-on-arp
FastIron SuperX Switch(config-ve-3)# exit
FastIron SuperX Switch(config-vlan-4)# interface ve 4
FastIron SuperX Switch(config-ve-4)# ip follow ve 2
FastIron SuperX Switch(config-ve-4)# ip use-acl-on-arp
FastIron SuperX Switch(config-ve-4)# exit
 [no] ip use-acl-on-arp [ <access-list-number> ]
```

When the **use-acl-on-arp** command is configured, the ARP module checks the source IP address of the ARP request packets received on the interface. It then applies the specified ACL policies to the packet. Only the packet with the IP address that the ACL permits will be allowed to be to be written in the ARP table; those that are not permitted will be dropped.

The <access-list-number> parameter identifies the ID of the standard ACL that will be used to filter the packet. Only the source and destination IP addresses will be used to filter the ARP packet. You can do one of the following for <access-list-number>:

- Enter an ACL ID to explicitly specify the ACL to be used for filtering. In the example above, the line FastIron SuperX Switch `(config-ve-2)# ip use-acl-on-arp 103` specifies ACL 103 to be used as the filter.
- Allow the ACL ID to be inherited from the IP ACLs that have been defined for the device. In the example above, the line FastIron SuperX Switch `(config-ve-4)# ip use-acl-on-arp` allows the ACL to be inherited from IP ACL 101 because of the ip follow relationship between virtual routing interface 2 and virtual routing interface 4. Virtual routing interface 2 is configured with IP ACL 101; thus virtual routing interface 4 inherits IP ACL 101.

ARP requests will not be filtered by ACLs if one of the following conditions occur:

- If the ACL is to be inherited from an IP ACL, but there is no IP ACL defined.
- An ACL ID is specified for the **use-acl-on-arp** command, but no IP address or “any any” filtering criteria have been defined under the ACL ID.

Displaying ACL Filters for ARP

To determine which ACLs have been configured to filter ARP requests, enter a command such as the following:

```
FastIron SuperX Switch(config)# show acl-on-arp

Port ACL ID Filter Count

2 103 10

3 102 23

4 101 12
```

Syntax: show acl-on-arp [ethernet [<slotnum>/<portnum>] | loopback [<num>] | ve [<num>]]

If the slot number (chassis devices only) and interface number is not specified, all ports on the device that use ACLs for ARP filtering will be included in the display.

The Filter Count column shows how many ARP packets have been dropped on the interface since the last time the count was cleared.

Clearing Filter Count

To clear the filter count for all interfaces on the device, enter a command such as the following:

```
FastIron SuperX Switch(config)# clear acl-on-arp
```

Syntax: clear acl-on-arp

The command resets the filter count on all interfaces in a device back to zero

Configuring the BOOTP/DHCP Reply Source Address

You can configure the Foundry device so that a BOOTP/DHCP reply to a client contains the server’s IP address as the source address instead of the router’s IP address. To do so, enter the following command at the Global CONFIG level of the CLI:

```
FastIron SuperX Switch(config)# ip helper-use-responder-ip
```

Syntax: [no] ip helper-use-responder-ip

VSRP-Aware Security

Software release 03.0.00 enhances the security of VSRP-aware switches against unauthorized VSRP hello packets by enabling you to configure VSRP-aware security parameters.

Without VSRP-aware security configured, a VSRP-aware device passively learns the authentication method conveyed by the received VSRP hello packet. The VSRP-aware device then stores the authentication method until it ages out with the aware entry.

With VSRP-aware security, you can:

- Define the specific authentication parameters that a VSRP-aware device will use on a VSRP backup switch. The authentication parameters that you define will not age out.
- Define a list of ports that have authentic VSRP backup switch connections. For ports included in the list, the VSRP-aware switch will process VSRP hello packets using the VSRP-aware security configuration. Conversely, for ports not included in the list, the VSRP-aware switch will not use the VSRP-aware security configuration.

If VSRP hello packets do not meet the acceptance criteria, the VSRP-aware device forwards the packets normally, without any VSRP-aware security processing.

Specifying an Authentication String for VSRP Hello Packets

The following configuration defines **pri-key** as the authentication string for accepting incoming VSRP hello packets. In this example, the VSRP-aware device will accept all incoming packets that have this authorization string.

```
FastIron SuperX Router(config)# vlan 10
FastIron SuperX Router(config-vlan-10)# vsrp-aware vrid 3 simple-text-auth pri-key
```

Syntax: vsrp-aware vrid <vrid number> simple text auth <string>

Specifying no Authentication for VSRP Hello Packets

The following configuration specifies no authentication as the preferred VSRP-aware security method. In this case, the VSRP device will not accept incoming packets that have authentication strings.

```
FastIron SuperX Router(config)# vlan 10
FastIron SuperX Router(config-vlan-10)# vsrp-aware vrid 2 no-auth
```

Syntax: vsrp-aware vrid <vrid number> no-auth

The following configuration specifies no authentication for VSRP hello packets received on ports 1/1, 1/2, 1/3, and 1/4 in VRID 4. For these ports, the VSRP device will not accept incoming packets that have authentication strings.

```
FastIron SuperX Router(config)# vlan 10
FastIron SuperX Router(config-vlan-10)# vsrp-aware vrid 4 no-auth port-list ethe 1/
1 to 1/4
```

Syntax: vsrp-aware vrid <vrid number> no-auth port-list <port range>

<vrid number> is a valid VRID (from 1 to 255).

no-auth specifies no authentication as the preferred VSRP-aware security method. The VSRP device will not accept incoming packets that have authentication strings.

simple-text-auth <string> specifies the authentication string for accepting VSRP hello packets, where <string> can be up to 8 characters.

port-list <port range> specifies the range of ports to include in the configuration.

Dynamic ARP Inspection

Dynamic ARP Inspection (DAI) enables the Foundry device to intercept and examine all ARP request and response packets in a subnet and discard those packets with invalid IP to MAC address bindings. DAI can prevent common man-in-the-middle (MiM) attacks such as ARP cache poisoning, and disallow misconfiguration of client IP addresses.

ARP Poisoning

ARP provides IP communication within a Layer 2 broadcast domain by mapping an IP address to a MAC address. Before a host can talk to another host, it must map the IP address to a MAC address first. If the host does not have the mapping in its ARP table, it creates an ARP request to resolve the mapping. All computers on the subnet will receive and process the ARP requests, and the host whose IP address matches the IP address in the request will send an ARP reply.

An ARP poisoning attack can target hosts, switches, and routers connected to the Layer 2 network by poisoning the ARP caches of systems connected to the subnet and by intercepting traffic intended for other hosts on the subnet. For instance, a malicious host can reply to an ARP request with its own MAC address, thereby causing other hosts on the same subnet to store this information in their ARP tables or replace the existing ARP entry. Furthermore, a host can send gratuitous replies without having received any ARP requests. A malicious host can also send out ARP packets claiming to have an IP address that actually belongs to another host (e.g. the default router). After the attack, all traffic from the device under attack flows through the attacker's computer and then to the router, switch, or host.

How DAI Works

DAI allows only valid ARP requests and responses to be forwarded.

A Foundry device on which DAI is configured does the following:

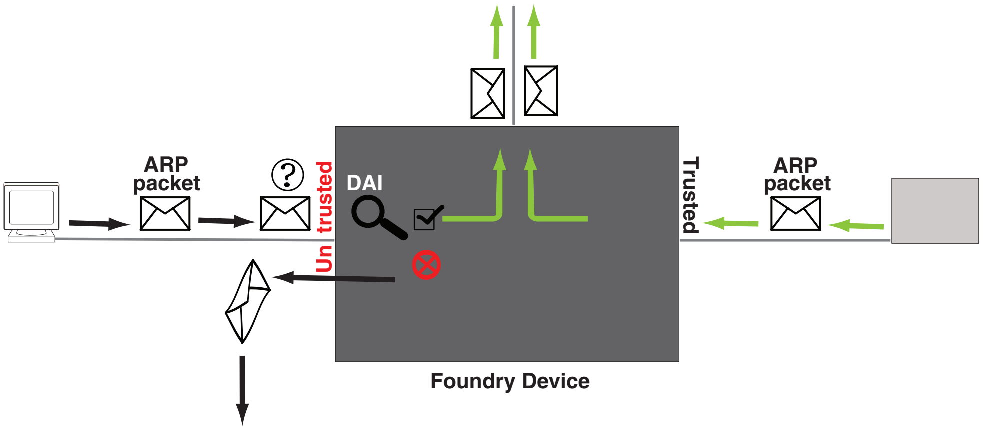
- Intercepts ARP packets received by the system CPU
- Inspects all ARP requests and responses received on untrusted ports
- Verifies that each of the intercepted packets has a valid IP-to-MAC address binding before updating the local ARP table, or before forwarding the packet to the appropriate destination
- Drops invalid ARP packets

When you enable DAI on a VLAN, by default, all member ports are untrusted. You must manually configure trusted ports. In a typical network configuration, ports connected to host ports are untrusted. You configure ports connected to other switches or routers as trusted.

DAI inspects ARP packets received on untrusted ports, as shown in Figure 9. DAI carries out the inspection based on IP-to-MAC address bindings stored in a trusted binding database. For the Foundry device, the binding database is the ARP table, which supports DAI, DHCP snooping, and IP Source Guard. To inspect an ARP request packet, DAI checks the source IP and source MAC address against the ARP table. For an ARP reply packet, DAI checks the source IP, source MAC, destination IP, and destination MAC addresses. DAI forwards the valid packets and discards those with invalid IP-to-MAC address bindings.

When ARP packets reach a trusted port, DAI lets them through, as shown in Figure 9.

Figure 9 Dynamic ARP Inspection at Work



ARP Entries

DAI uses the IP/MAC mappings in the ARP table to validate ARP packets received on untrusted ports.

ARP entries in the ARP table derive from the following:

- Dynamic ARP – normal ARP learned from trusted ports.
- Static ARP – statically configured IP/MAC/port mapping.
- Inspection ARP – statically configured IP/MAC mapping, where the port is initially unspecified. The actual physical port mapping will be resolved and updated from validated ARP packets. See "Configuring an Inspection ARP Entry" on page 54.
- DHCP-Snooping ARP – information collected from snooping DHCP packets when DHCP snooping is enabled on VLANs.

The status of an ARP entry is either pending or valid:

- Valid – the mapping is valid, and the port is resolved. This is always the case for static ARP entries.
- Pending – for normal dynamic, inspection ARP, and DHCP-Snooping ARP entries before they are resolved, and the port mapped. Their status changes to valid when they are resolved, and the port mapped.

See also: "System Reboot and the Binding Database" on page 58.

Support for DAI on Trunk Ports

When the primary port of a trunk group is DAI trusted, secondary ports of the trunk group inherit the trust behavior. Foundry recommends that trunk ports be trusted for DAI.

Configuring DAI

Configuring DAI consists of the following steps:

1. Configure inspection ARP entries for hosts on untrusted ports. See "Configuring an Inspection ARP Entry" on page 54.
2. Enable DAI on a VLAN to inspect ARP packets. See "Enabling DAI on a VLAN" on page 55.
3. Configure the trust settings of the VLAN members. ARP packets received on *trusted* ports bypass the DAI validation process. ARP packets received on *untrusted* ports go through the DAI validation process. See "Enabling Trust on a Port" on page 55.
4. Enable DHCP snooping to populate the DHCP snooping IP-to-MAC binding database.

The following shows the default settings of DAI:

Feature	**Default**
Dynamic ARP Inspection	Disabled
Trust setting for ports	Untrusted

Configuring an Inspection ARP Entry

Static ARP and static inspection ARP entries need to be configured for hosts on untrusted ports. Otherwise, when DAI checks ARP packets from these hosts against entries in the ARP table, it will not find any entries for them, and the Foundry device will not allow and learn ARP from an untrusted host.

When the inspection ARP entry is resolved with the correct IP/MAC mapping, its status changes from pending to valid.

To configure an inspection ARP entry, enter commands such as the following:

```
FastIron SuperX Switch(config)# interface ve 2
FastIron SuperX Switch(config-vif-2)# ip address 20.20.20.1/24
FastIron SuperX Switch(config-vif-2)# exit
FastIron SuperX Switch(config)# arp 1  20.20.20.12  0001.0002.0003 inspection
```

The commands change the CLI to the interface configuration level for virtual interface 2, assign it an IP address, and then define an inspection ARP entry, mapping a device's IP address 20.20.20.12 with its MAC address 0001.0002.0003.

Syntax: [no] arp <index> <ip-addr> <mac-addr> inspection

The index can be from 1 up to the maximum number of static entries allowed.

The <ip-addr> <mac-addr> parameter specifies a device's IP address and MAC address pairing.

Enabling DAI on a VLAN

DAI is disabled by default. To enable DAI on an existing VLAN, enter the following command:

```
FastIron SuperX Switch(config)# ip arp inspection vlan 2
```

The command enables DAI on VLAN 2. ARP packets from untrusted ports in VLAN 2 will undergo DAI inspection.

Syntax: [no] ip arp inspection vlan <vlan-number>

The <vlan-number> variable specifies the ID of a configured VLAN.

Enabling Trust on a Port

The default trust setting for a port is untrusted. For ports that are connected to host ports, leave their trust settings as untrusted.

To enable trust on a port, enter commands such as the following:

```
FastIron SuperX Switch(config)# interface ethernet 1/4
FastIron SuperX Switch(config-if-e10000-1/4)# arp inspection trust
```

The commands change the CLI to the interface configuration level of port 1/4 and set the trust setting of port 1/4 to trusted.

Syntax: [no] arp inspection trust

Displaying ARP Inspection Status and Ports

To display the ARP inspection status for a VLAN and the trusted/untrusted ports in the VLAN, enter the following command:

```
FastIron SuperX Switch# show ip arp inspection vlan 2

IP ARP inspection VLAN 2: Disabled
  Trusted Ports :   ethe 1/4
  Untrusted Ports : ethe 2/1 to 2/3 ethe 4/1 to 4/24 ethe 6/1 to 6/4 ethe 8/1 to
 8/4
```

Syntax: show ip arp inspection [vlan <vlan_id>]

The <vlan_id> variable specifies the ID of a configured VLAN.

Displaying the ARP Table

To display the ARP table, enter the following command:

```
FastIron SuperX Switch# show arp
    IP Address       Mac Address    Type    Age Port       Status Vlan

 1        100.1.1.1   000b.cd3d.80e6 Inspect   1 1/1         Valid    1
 2         22.2.2.2   0001.0002.0003 Inspect   0 ----        Pend ----
 3        100.1.1.1   000b.cd3d.80e6 Dynamic   1 1/1         Valid    1
 4         55.5.5.5   0005.0005.0005 DHCP      1  ---        Pend  ----
 5         55.5.5.6   0006.0006.0006 Dy-DHCP   1  2/2        Valid  22
```

The command displays all ARP entries in the system.

Syntax: show arp

The display shows the following information. The number in the left column of the CLI display is the row number of the entry in the ARP cache. This number is not related to the number you assign to static MAC entries in the static ARP table.

Table 9: CLI Display of ARP Cache

This Field...	Displays...
IP Address	The IP address of the device.
MAC Address	The MAC address of the device.
Type	The ARP type, which can be one of the following: • Dynamic – The Layer 3 Switch learned the entry from an incoming packet on a trusted port. • Static – The Layer 3 Switch loaded the entry from the static ARP table when the device for the entry was connected to the Layer 3 Switch. • Inspect (Inspection ARP) – The Layer 3 Switch learned the entry from a statically configured IP/MAC mapping, where the port was initially unspecified. • DHCP (DHCP-Snooping ARP) – The Layer 3 Switch learned the entry from DHCP. • Dy-DHCP (Dynamic + DHCP-Snooping ARP) – The Layer 3 Switch learned the entry from an incoming packet on a trusted port and from DHCP.
Age	The number of minutes the entry has remained unused. If this value reaches the ARP aging period, the entry is removed from the table. **Note**: Static entries do not age out.
Port	This field shows one of the following: • The port on which the entry was learned. • ---- indicates that a port was not specified.

Table 9: CLI Display of ARP Cache (Continued)

This Field...	Displays...
Status	The status, which can be one of the following: • Valid – The ARP entry was resolved with the correct IP/MAC mapping. Static ARP entries are always valid. • Pend – The ARP entry is not yet resolved.
VLAN	This field can be one of the following: • The ID of the configured VLAN • ---- indicates that a VLAN was not specified.

DHCP Snooping

Dynamic Host Configuration Protocol (DHCP) snooping enables the Foundry device to filter untrusted DHCP packets in a subnet. DHCP snooping can ward off MiM attacks, such as a malicious user posing as a DHCP server sending false DHCP server reply packets with the intention of misdirecting other users. DHCP snooping can also stop unauthorized DHCP servers and prevent errors due to user misconfiguration of DHCP servers.

Often DHCP snooping is used together with Dynamic ARP Inspection and IP Source Guard.

How DHCP Snooping Works

When enabled on a VLAN, DHCP snooping stands between untrusted ports (those connected to host ports) and trusted ports (those connected to DHCP servers). A VLAN with DHCP snooping enabled forwards DHCP request packets from clients and discards DHCP server reply packets on untrusted ports, and it forwards DHCP server reply packets on trusted ports to DHCP clients, as shown in the following figures:

Figure 10 DHCP Snooping at Work - on Untrusted Port

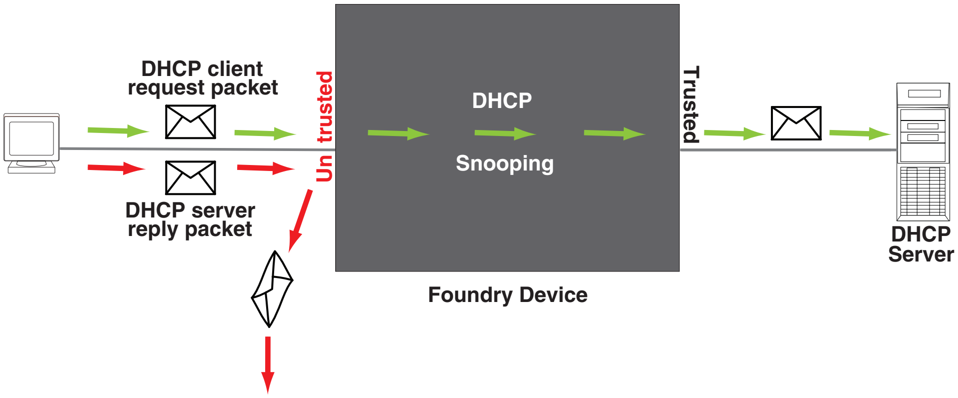
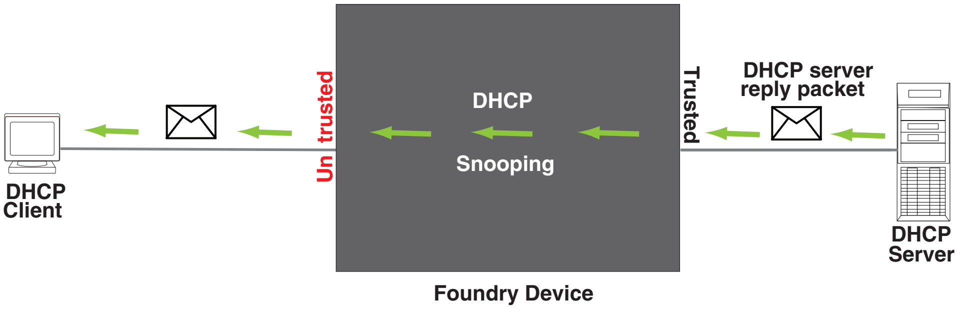


Figure 11 DHCP Snooping at Work - on Trusted Port



DHCP Binding Database

When it forwards DHCP server reply packets on trusted ports, the Foundry device saves the client IP-to-MAC address binding information in the DHCP binding database. This is how the DHCP snooping binding table is populated. The information saved includes MAC address, IP address, lease time, VLAN number, and port number.

In the Foundry device, the DHCP binding database is integrated with the enhanced ARP table, which is used by Dynamic ARP Inspection. For more information, see “ARP Entries” on page 54.

The lease time will be refreshed when the client renews its IP address with the DHCP server; otherwise the Foundry device removes the entry when the lease time expires.

System Reboot and the Binding Database

To allow DAI and DHCP snooping to work smoothly across a system reboot, the binding database is saved to a file in the system flash memory after an update to the binding database, with a 30 second delay. The flash file is written and read only if DHCP snooping is enabled.

Support for DHCP Snooping on Trunk Ports

When the primary port of a trunk group is DHCP-snooping trusted, secondary ports of the trunk group inherit the trust behavior. Foundry recommends that trunk ports be trusted for DHCP snooping.

Configuring DHCP Snooping

Configuring DHCP snooping consists of the following steps:

1. Enable DHCP snooping on a VLAN. See “Enabling DHCP Snooping on a VLAN” on page 58.
2. For ports that are connected to a DHCP server, change their trust setting to trusted. See “Enabling Trust on a Port” on page 58.

The following shows the default settings of DHCP snooping:

Feature	Default
DHCP snooping	Disabled
Trust setting for ports	Untrusted

Enabling DHCP Snooping on a VLAN

DHCP packets for a VLAN with DHCP snooping enabled are inspected.

DHCP snooping is disabled by default. To enable DHCP snooping for a VLAN, enter the following global command:

```
FastIron SuperX Switch(config)# ip dhcp snooping vlan 2
```

The command enables DHCP snooping on VLAN 2.

Syntax: [no] ip dhcp snooping vlan <vlan-number>

The <vlan-number> variable specifies the ID of a configured VLAN.

Enabling Trust on a Port

The default trust setting for a port is untrusted. To enable trust on a port connected to a DHCP server, enter commands such as the following:

```
FastIron SuperX Switch(config)# interface ethernet 1/1
FastIron SuperX Switch(config-if-e10000-1/1)# dhcp snooping trust
```

Port 1/1 is connected to a DHCP server. The commands change the CLI to the interface configuration level of port 1/1 and set the trust setting of port 1/1 to trusted.

Syntax: [no] dhcp snooping trust

Clearing the DHCP Binding Database

You can clear the DHCP binding database using the CLI command **clear DHCP**. You can remove all entries in the database, or remove entries for a specific IP address only.

To remove all entries from the DHCP binding database, enter the following command:

```
FastIron SuperX Switch(config)# clear dhcp
```

To clear entries for a specific IP address, enter a command such as the following:

```
FastIron SuperX Switch(config)# clear dhcp 10.10.102.4
```

Syntax: clear dhcp [<IP-address>]

Displaying DHCP Snooping Status and Ports

To display the DHCP snooping status for a VLAN and the trusted/untrusted ports in the VLAN, enter the following command:

```
 FastIron SuperX Switch(config)# show ip dhcp snooping vlan 2

 IP DHCP snooping VLAN 2: Enabled
```

Syntax: show ip dhcp snooping [vlan <vlan-id>]

Displaying DHCP Binding Entry and Status

To display the DHCP binding entry and its current status, use the **show arp** command.

DHCP Snooping Configuration Example

The following example configures VLAN 2 and VLAN 20, and changes the CLI to the global configuration level to enable DHCP snooping on the two VLANs. The commands are as follows:

```
FastIron SuperX Switch(config)# vlan 2
FastIron SuperX Switch(config-vlan-2)# untagged ethe 1/3 to 1/4
FastIron SuperX Switch(config-vlan-2)# router-interface ve 2
FastIron SuperX Switch(config-vlan-2)# exit
FastIron SuperX Switch(config)# ip dhcp snooping vlan 2

FastIron SuperX Switch(config)# vlan 20
FastIron SuperX Switch(config-vlan-20)# untagged ethe 1/1 to 1/2
FastIron SuperX Switch(config-vlan-20)# router-interface ve 20
FastIron SuperX Switch(config-vlan-20)# exit
FastIron SuperX Switch(config)# ip dhcp snooping vlan 20
```

On VLAN 2, client ports 1/3 and 1/4 are untrusted by default: all client ports are untrusted. Hence, only DHCP client request packets received on ports 1/3 and 1/4 are forwarded.

On VLAN 20, ports 1/1 and 1/2 are connected to a DHCP server. DHCP server ports are set to trusted:

```
FastIron SuperX Switch(config)# interface ethernet 1/1
FastIron SuperX Switch(config-if-e10000-1/1)# dhcp snooping trust
FastIron SuperX Switch(config-if-e10000-1/1)# exit
FastIron SuperX Switch(config)# interface ethernet 1/2
FastIron SuperX Switch(config-if-e10000-1/2)# dhcp snooping trust
FastIron SuperX Switch(config-if-e10000-1/2)# exit
```

Hence, DHCP sever reply packets received on ports 1/1 and 1/2 are forwarded, and client IP/MAC binding information is collected.

The example also sets the DHCP server address for the local relay agent:

```
FastIron SuperX Switch(config)# interface ve 2
FastIron SuperX Switch(config-vif-2)# ip address 20.20.20.1/24
FastIron SuperX Switch(config-vif-2)# ip helper-address 30.30.30.4
FastIron SuperX Switch(config-vif-2)# interface ve 20
FastIron SuperX Switch(config-vif-20)# ip address 30.30.30.1/24
```

IP Source Guard

You can use IP Source Guard together with Dynamic ARP Inspection on untrusted ports. See “DHCP Snooping” on page 57 and “Dynamic ARP Inspection” on page 52.

Foundry’s implementation of the IP Source Guard feature supports configuration on a port, on specific VLAN members on a port (Layer 2 devices only), and on specific ports on a virtual interface (VE) (Layer 3 device only).

When IP Source Guard is first enabled, only DHCP packets are allowed and all other IP traffic is blocked. When the system learns a valid IP address, IP Source Guard then allows IP traffic. Only the traffic with valid source IP addresses are permitted. The system learns of a valid IP address from DHCP Snooping. When it learns a valid IP address, the system permits the learned source IP address.

When a new IP source entry binding on the port is created or deleted, the ACL will be recalculated and reapplied in hardware to reflect the change in IP source binding. By default, if IP Source Guard is enabled without any IP source binding on the port, an ACL that denies all IP traffic is loaded on the port.

Configuration Notes and Feature Limitations

- Foundry devices support IP Source Guard together with user ACLs (similar to ACLs for Dot1x).

NOTE: If you are configuring IP Source Guard on a port that belongs to more than one VLAN, you must first enable per-port-per-VLAN ACLs (ACL filtering based on VLAN membership or VE port membership). To enable this feature, enter the following command at the Global CONFIG Level of the CLI:

```
FESX424 Switch (config)# enable acl-per-port-per-vlan
FESX424 Switch (config)# write memory
FESX424 Switch (config)# exit
FESX424 Switch# reload
```

NOTE: You must save the configuration and reload the software to place the change into effect.

- The following limitations apply when configuring IP Source Guard on Layer 3 devices:
  - You cannot enable IP Source Guard on a tagged port on a Layer 3 device. To enable IP Source Guard on a tagged port, enable it on a per-VE basis.
  - You cannot enable IP Source Guard on an untagged port with VE on a Layer 3 device. To enable IP Source Guard in this configuration, enable it on a per-VE basis.
- There are no restrictions for Layer 2, either on the port or per-VLAN.
- You cannot enable IP Source Guard on a port that has any of the following features enabled:
  - MAC address filter
  - Rate limiting
  - Trunk port
  - 802.1x with ACLs
  - PBR
- A port on which IP Source Guard is enabled limits the support of IP addresses, VLANs, and ACL rules per port. An IP Source Guard ports supports a maximum of:
  - 64 IP addresses
  - 64 VLANs
  - 64 rules per ACL

Enabling IP Source Guard on a Port

You enable IP Source Guard on DHCP snooping untrusted ports. See “DHCP Snooping” on page 57 for how to configure DHCP and DHCP untrusted ports.

By default, IP Source Guide is disabled. To enable IP Source Guard on a DHCP untrusted port, enter the following commands:

```
FastIron SuperX Switch(config)# interface ethernet 1/4
FastIron SuperX Switch(config-if-e10000-1/4)# source-guard enable
```

The commands change the CLI to the interface configuration level for port 1/4 and enable IP Source Guard on the port.

Syntax: [no] source-guard enable

Defining Static IP Source Bindings

You can manually enter valid IP addresses in the binding database. To do so, enter a command such as the following:

```
FastIron SuperX Switch(config)# ip source binding 10.10.10.1 e 2/4 vlan 4
```

Syntax: [no] ip source binding <IP-address> ethernet [<slotnum>/]<portnum> [vlan <vlannum>]

For <IP-address>, enter a valid IP address.

For ethernet [<slotnum>/]<portnum>, enter a valid port number. <slotnum> is required on chassis devices only.

[vlan <vlannum>] is required If Source Guard is enabled on a PPP VLAN or PPP virtual interface. Otherwise, this parameter is optional. If you enter a VLAN number, the binding applies to that VLAN only. If you do not enter a VLAN number, the static binding applies to all VLANs associated with the port. Note that since static IP source bindings consume system resources, you should avoid unnecessary bindings.

Enabling IP Source Guard Per-Port-Per-VLAN

To enable IP Source Guard per-port-per VLAN, enter commands such as the following:

```
FESX424 Switch(config)# vlan 12 name vlan12
FESX424 Switch(config-vlan-12)# untag ethernet 5 to 8
FESX424 Switch(config-vlan-12)# tag ethernet 23 to 24
FESX424 Switch(config-vlan-12)#exit
FESX424 Switch(config)# int e 23
FESX424 Switch(config-if-e1000-23)# per-vlan vlan12
FESX424 Switch(config-if-e1000-23-vlan-12))# source-guard enable
```

The commands in this example configure port-based VLAN 12, and add ports e 5 – 8 as untagged ports and ports e 23 – 24 as tagged ports to the VLAN.   The last two commands enable IP Source Guard on port e 23, a member of VLAN 12.

Syntax: [no] source-guard enable

Enabling IP Source Guard on a VE

To enable IP Source Guard on a virtual interface, enter commands such as the following:

```
FESX424F+2XG Router(config)#vlan 2
FESX424F+2XG Router(config-vlan-2)#tag e1
Added tagged port(s) ethe 1 to port-vlan 2
FESX424F+2XG Router(config-vlan-2)#router-int ve 2
FESX424F+2XG Router(config-vlan-2)#int ve 2
FESX424F+2XG Router(config-vif-2)#source-guard enable e 1
```

Syntax: [no] source-guard enable

Displaying Learned IP Addresses

To display the learned IP addresses for IP Source Guard ports, use the **show ip source-guard** command.

```
SW-FESX424 Switch(config)# sh ip source-guard
  Interface  Flter-type  Flter-mode       IP-address        Mac-address     Vlan
  ---------  ----------  ----------       ----------        -----------     ----
1                 ip       Inactive            1.1.1.1                     none
1                 ip         active         64.12.12.5                     22
1                 ip       Inactive            4.4.4.6                     none
1                 ip         active            5.5.5.5                     none
1                 ip       Inactive            4.4.4.2                     23
1                 ip       Inactive            4.4.4.3                     none
```

Syntax: show ip source-guard [ethernet [<slotnum/>]<portnum> vlan <vlan-ID>]

<slotnum> is required on chassis devices.

Table 10: CLI Display of ARP Cache

This Field...	Displays...
Interface	The interface number
Filter-type	In release 03.0.01, the filter type is IP.
Filter-mode	The filter mode can be one of the following: • Inactive – The entry is not applied to the interface. • Active – The entry is active on the interface.
IP-address	The IP address of the device.
Mac-address	The MAC address of the device. This field applies only when the Filter-type is MAC. **NOTE:** This field is reserved for future use.
VLAN	This field can be one of the following: • The ID of the configured VLAN • None – A VLAN was not specified for static binding

SSHv2

Secure Shell (SSH) is a mechanism for allowing secure remote access to management functions on a Foundry device. SSH provides a function similar to Telnet. Users can log into and configure the device using a publicly or commercially available SSH client program, just as they can with Telnet. However, unlike Telnet, which provides no security, SSH provides a secure, encrypted connection to the device.

SSH v2 is supported on the Foundry device. Foundry's SSHv2 implementation is compatible with all versions of the SSHv2 protocol (2.1, 2.2, and so on). At the beginning of an SSH session, the Foundry device negotiates the version of SSHv2 to be used. The highest version of SSHv2 supported by both the Foundry device and the client is the version that is used for the session. Once the SSHv2 version is negotiated, the encryption algorithm with the highest security ranking is selected to be used for the session.

Releases prior to 03.0.00 support SSH version 1 (SSHv1). Starting with release 03.0.00, SSH version 2 (SSHv2) is supported and SSHv1 is no longer supported.

Also, Foundry devices support Secure Copy (SCP) for securely transferring files between a Foundry device and SCP-enabled remote hosts.

SSH Version 2 Support

SSHv2 is a substantial revision of Secure Shell, comprising the following hybrid protocols and definitions:

- SSH Transport Layer Protocol
- SSH Authentication Protocol
- SSH Connection Protocol
- SECSH Public Key File Format
- SSH Fingerprint Format
- SSH Protocol Assigned Numbers
- SSH Transport Layer Encryption Modes
- SCP/SFTP/SSH URI Format

If you are using redundant management modules, you can synchronize the DSA host key pair between the active and standby modules by entering the **sync-standby** command at the Privileged EXEC level of the CLI.

Tested SSHv2 Clients

The following SSH clients have been tested with SSHv2:

- SSH Secure Shell 3.2.3
- Van Dyke SecureCRT 4.0 and 4.1
- F-Secure SSH Client 5.3 and 6.0
- PuTTY 0.54 and 0.56
- OpenSSH 3.5_p1 and 3.6.1p2
- Solaris Sun-SSH-1.0

Supported Encryption Algorithms for SSHv2

3DES is the encryption algorithms supported in Foundry’s implementation of SSHv2.

Supported MAC (Message Authentication Code) Algorithms

SHA 1 is the MAC algorithm supported in Foundry’s implementation of SSHv2:

Configuring SSH

Foundry’s implementation of SSH supports two kinds of user authentication:

- ***DSA challenge-response authentication***, where a collection of public keys are stored on the device. Only clients with a private key that corresponds to one of the stored public keys can gain access to the device using SSH.
- ***Password authentication***, where users attempting to gain access to the device using an SSH client are authenticated with passwords stored on the device or on a TACACS/TACACS+ or RADIUS server

Both kinds of user authentication are enabled by default. You can configure the device to use one or both of them.

To configure Secure Shell on a Foundry device, do the following:

1. Generate a host DSA public and private key pair for the device.
2. Configure DSA challenge-response authentication.
3. Set optional parameters.

You can also view information about active SSH connections on the device as well as terminate them.

Generating a Host Key Pair

When SSH is configured, a public and private ***host DSA key pair*** is generated for the Foundry device. The SSH server on the Foundry device uses this host DSA key pair, along with a dynamically generated ***server DSA key pair***, to negotiate a session key and encryption method with the client trying to connect to it.

The host DSA key pair is stored in the Foundry device's system-config file. Only the public key is readable. The public key should be added to a "known hosts" file (for example, $HOME/.ssh/known_hosts on UNIX systems) on the clients who want to access the device. Some SSH client programs add the public key to the known hosts file automatically; in other cases, you must manually create a known hosts file and place the Foundry device's public key in it.

While the SSH listener exists at all times, sessions can't be started from clients until a key is generated. Once a key is generated, clients can start sessions. The keys are also not displayed in the configuration file by default. To display the keys, use the **ssh show-host-keys** command in Privileged EXEC mode. To generate a public and private DSA host key pair on a Foundry device, enter the following commands:

```
FastIron SuperX Switch(config)# crypto key generate
```

When a host key pair is generated, it is saved to the flash memory of all management modules.

To disable SSH in SSHv2 on a Foundry device, enter the following commands:

```
FastIron SuperX Switch(config)# crypto key zeroize
```

When SSH is disabled, it is deleted from the flash memory of all management modules.

Syntax: crypto key generate | zeroize

The **generate** keyword places an DSA host key pair in the flash memory and enables SSH on the device.

The **zeroize** keyword deletes the DSA host key pair from the flash memory and disables SSH on the device.

By default, public keys are hidden in the running configuration. You can optionally configure the Foundry device to display the DSA host key pair in the running configuration file entering the following command:

```
FastIron SuperX Switch# ssh show-host-keys
```

Syntax: ssh show-host-keys

To hide the public keys in the running configuration file, enter the following command:

```
FastIron SuperX Switch# ssh no-show-host-keys
```

Syntax: ssh no-show-host-keys

Providing the Public Key to Clients

If you are using SSH to connect to a Foundry device from a UNIX system, you may need to add the Foundry device's public key to a "known hosts" file; for example, $HOME/.ssh/known_hosts. The following is an example of an entry in a known hosts file:

```
AAAAB3NzaC1kc3MAAACBAPY8ZOHY2yFSJA6XYC9HRwNHxaehvx5wOJ0rzZdzoSOXxbET W6ToHv8D1UJ/
z+zHo9Fiko5XybZnDIaBDHtblQ+Yp7StxyltHnXF1YLfKD1G4T6JYrdH YI14Om
1eg9e4NnCRleaqoZPF3UGfZia6bXrGTQf3gJq2e7Yisk/gF+1VAAAAFQDb8D5cv
wHWTZDPfX0D2s9Rd7NBvQAAAIEAlN92+Bb7D4KLYk3IwRbXblwXdkPggA4pfdtW9v
GfJ0/RHd+NjB4eo1D+0dix6tXwYGN7PKS5R/FXPNwxHPapcj9uL1Jn2AWQ2dsknf+i/FAA
vioUPkmdMc0zuWoSOEsSNhVDtX3WdvVcGcBq9cetzrtOKWOocJmJ80qadxTRHtUAAACB
AN7CY+KKv1gHpRzFwdQm7HK9bb1LAo2KwaoXnadFgeptNBQeSXG1vO+JsvphVMBJc9HS
n24VYtYtsMu74qXviYjziVucWKjjKEb11juqnF0GDlB3VVmxHLmxnAz643WK42Z7dLM5
sY29ouezv4Xz2PuMch5VGPP+CDqzCM4loWgV
```

Configuring DSA Challenge-Response Authentication

With DSA challenge-response authentication, a collection of clients' public keys are stored on the Foundry device. Clients are authenticated using these stored public keys. Only clients that have a private key that corresponds to one of the stored public keys can gain access to the device using SSH.

When DSA challenge-response authentication is enabled, the following events occur when a client attempts to gain access to the device using SSH:

1. The client sends its public key to the Foundry device.
2. TheFoundry device compares the client's public key to those stored in memory.
3. If there is a match, the Foundry device uses the public key to encrypt a random sequence of bytes.
4. The Foundry device sends these encrypted bytes to the client.
5. The client uses its private key to decrypt the bytes.
6. The client sends the decrypted bytes back to the Foundry device.
7. The Foundry device compares the decrypted bytes to the original bytes it sent to the client. If the two sets of bytes match, it means that the client's private key corresponds to an authorized public key, and the client is authenticated.

Setting up DSA challenge-response authentication consists of the following steps:

1. Importing authorized public keys into the Foundry device.
2. Enabling DSA challenge response authentication

Importing Authorized Public Keys into the Foundry device

SSH clients that support DSA authentication normally provide a utility to generate an DSA key pair. The private key is usually stored in a password-protected file on the local host; the public key is stored in another file and is not protected. You should collect one public key from each client to be granted access to the Foundry device and place all of these keys into one file. This public key file is imported into the Foundry device.

The following is an example of a public key file containing one public keys:

```
---- BEGIN SSH2 PUBLIC KEY ----
Comment: DSA Public Key
AAAAB3NzaC1kc3MAAACBAPY8ZOHY2yFSJA6XYC9HRwNHxaehvx5wOJ0rzZdzoSOXxbET W6ToHv8D1UJ/
z+zHo9Fiko5XybZnDIaBDHtblQ+Yp7StxyltHnXF1YLfKD1G4T6JYrdH YI14Om
1eg9e4NnCRleaqoZPF3UGfZia6bXrGTQf3gJq2e7Yisk/gF+1VAAAAFQDb8D5cv
wHWTZDPfX0D2s9Rd7NBvQAAAIEAlN92+Bb7D4KLYk3IwRbXblwXdkPggA4pfdtW9v
GfJ0/RHd+NjB4eo1D+0dix6tXwYGN7PKS5R/FXPNwxHPapcj9uL1Jn2AWQ2dsknf+i/FAA
vioUPkmdMc0zuWoSOEsSNhVDtX3WdvVcGcBq9cetzrtOKWOocJmJ80qadxTRHtUAAACB
AN7CY+KKv1gHpRzFwdQm7HK9bb1LAo2KwaoXnadFgeptNBQeSXG1vO+JsvphVMBJc9HS
n24VYtYtsMu74qXviYjziVucWKjjKEb11juqnF0GDlB3VVmxHLmxnAz643WK42Z7dLM5
sY29ouezv4Xz2PuMch5VGPP+CDqzCM4loWgV
---- END SSH2 PUBLIC KEY ----
```

You can import the authorized public keys into the active configuration by loading them from a file on a TFTP server and are saved on the EEPROM of the chassis. If you import a public key file from a TFTP server, the file is automatically loaded into the active configuration the next time the device is booted.

To cause a public key file called pkeys.txt to be loaded from a TFTP server each time the Foundry device is booted, enter a command such as the following:

```
FastIron SuperX Switch(config)# ip ssh pub-key-file tftp 192.168.1.234 pkeys.txt
```

Syntax: ip ssh pub-key-file tftp | <tftp-server-ip-addr> <filename> [remove]

The <tftp-server-ip-addr> variable is the IP address of the tftp server that contains the public key file that you want to import into the Foundry device.

The <filename> variable is the name of the dsa public key file that you want to import into the Foundry device.

The **remove** parameter deletes the key from the system.

To display the currently loaded public keys, enter the following command:

```
FastIron SuperX Switch# show ip client-pub-key

---- BEGIN SSH2 PUBLIC KEY ----
Comment: DSA Public Key
AAAAB3NzaC1kc3MAAACBAPY8ZOHY2yFSJA6XYC9HRwNHxaehvx5wOJ0rzZdzoSOXxbET W6ToHv8D1UJ/
z+zHo9Fiko5XybZnDIaBDHtblQ+Yp7StxyltHnXF1YLfKD1G4T6JYrdH YI14Om
1eg9e4NnCRleaqoZPF3UGfZia6bXrGTQf3gJq2e7Yisk/gF+1VAAAAFQDb8D5cv
wHWTZDPfX0D2s9Rd7NBvQAAAIEAlN92+Bb7D4KLYk3IwRbXblwXdkPggA4pfdtW9v
GfJ0/RHd+NjB4eo1D+0dix6tXwYGN7PKS5R/FXPNwxHPapcj9uL1Jn2AWQ2dsknf+i/FAA
vioUPkmdMc0zuWoSOEsSNhVDtX3WdvVcGcBq9cetzrtOKWOocJmJ80qadxTRHtUAAACB
AN7CY+KKv1gHpRzFwdQm7HK9bb1LAo2KwaoXnadFgeptNBQeSXG1vO+JsvphVMBJc9HS
n24VYtYtsMu74qXviYjziVucWKjjKEb11juqnF0GDlB3VVmxHLmxnAz643WK42Z7dLM5
sY29ouezv4Xz2PuMch5VGPP+CDqzCM4loWgV
---- END SSH2 PUBLIC KEY ----
```

Syntax: show ip client-pub-key [| begin<expression> | exclude <expression> | include <expression>]

To clear the public keys from the buffers, enter the following command:

```
FastIron SuperX Switch# clear public-key
```

Syntax: clear public-key

Use the **ip ssh pub-key remove** command to delete the public key from the system.

Enabling DSA Challenge-Response Authentication

DSA challenge-response authentication is enabled by default. You can disable or re-enable it manually.

To enable DSA challenge-response authentication:

```
FastIron SuperX Switch(config)# ip ssh key-authentication yes
```

To disable DSA challenge-response authentication:

```
FastIron SuperX Switch(config)# ip ssh key-authentication no
```

Syntax: ip ssh key-authentication yes | no

Setting the Number of SSH Authentication Retries

By default, the Foundry device attempts to negotiate a connection with the connecting host three times. The number of authentication retries can be changed to between 1 – 5.

For example, the following command changes the number of authentication retries to 5:

```
FastIron SuperX Switch(config)# ip ssh authentication-retries 5
```

Syntax: ip ssh authentication-retries <number>

Deactivating User Authentication

After the SSH server on the Foundry device negotiates a session key and encryption method with the connecting client, user authentication takes place. Foundry's implementation of SSH supports DSA challenge-response authentication and password authentication.

With DSA challenge-response authentication, a collection of clients' public keys are stored on the Foundry device. Clients are authenticated using these stored public keys. Only clients that have a private key that corresponds to one of the stored public keys can gain access to the device using SSH.

With password authentication, users are prompted for a password when they attempt to log into the device (provided empty password logins are not allowed. If there is no user account that matches the user name and password supplied by the user, the user is not granted access.

You can deactivate one or both user authentication methods for SSH. Note that deactivating both authentication methods essentially disables the SSH server entirely.

To disable DSA challenge-response authentication:

```
FastIron SuperX Switch(config)# ip ssh key-authentication no
```

Syntax: ip ssh key-authentication yes | no

The default is “yes”.

To deactivate password authentication:

```
FastIron SuperX Switch(config)# ip ssh password-authentication no
```

Syntax: ip ssh password-authentication no | yes

The default is “yes”.

Enabling Empty Password Logins

By default, empty password logins are not allowed. This means that users with an SSH client are always prompted for a password when they log into the device. To gain access to the device, each user must have a user name and password. Without a user name and password, a user is not granted access.

If you enable empty password logins, users are ***not*** prompted for a password when they log in. Any user with an SSH client can log in without being prompted for a password.

To enable empty password logins:

```
FastIron SuperX Switch(config)# ip ssh permit-empty-passwd yes
```

Syntax: ip ssh permit-empty-passwd no | yes

Setting the SSH Port Number

By default, SSH traffic occurs on TCP port 22. You can change this port number. For example, the following command changes the SSH port number to 2200:

```
FastIron SuperX Switch(config)# ip ssh port 2200
```

Note that if you change the default SSH port number, you must configure SSH clients to connect to the new port. Also, you should be careful not to assign SSH to a port that is used by another service. If you change the SSH port number, Foundry recommends that you change it to a port number greater than 1024.

Syntax: ip ssh port <number>

Setting the SSH Login Timeout Value

When the SSH server attempts to negotiate a session key and encryption method with a connecting client, it waits a maximum of 120 seconds for a response from the client. If there is no response from the client after 120 seconds, the SSH server disconnects. You can change this timeout value to between 1 – 120 seconds. For example, to change the timeout value to 60 seconds:

```
FastIron SuperX Switch(config)# ip ssh timeout 60
```

Syntax: ip ssh timeout <seconds>

Designating an Interface as the Source for All SSH Packets

You can designate a loopback interface, virtual interface, or Ethernet port as the source for all SSH packets from the device. The software uses the IP address with the numerically lowest value configured on the port or interface as the source IP address for SSH packets originated by the device.

NOTE: When you specify a single SSH source, you can use only that source address to establish SSH management sessions with the Foundry device.

To specify the numerically lowest IP address configured on a loopback interface as the device's source for all SSH packets, enter commands such as a the following:

```
FastIron SuperX Switch(config)# int loopback 2
FastIron SuperX Switch(config-lbif-2)# ip address 10.0.0.2/24
FastIron SuperX Switch(config-lbif-2)# exit
FastIron SuperX Switch(config)# ip ssh source-interface loopback 2
```

The commands in this example configure loopback interface 2, assign IP address 10.0.0.2/24 to the interface, then designate the interface as the source for all SSH packets from the Foundry device.

Syntax: ip ssh source-interface ethernet <slot/port> | loopback <num> | ve <num>

The <num> parameter is a loopback interface or virtual interface number. The <slot/port> parameter specifies an ethernet port number. For example:

```
FastIron SuperX Switch(config)# interface ethernet 1/4
FastIron SuperX Switch(config-if-e10000-1/4)# ip address 209.157.22.110/24
FastIron SuperX Switch(config-if-e10000-1/4)# exit
FastIron SuperX Switch(config)# ip ssh source-interface ethernet 1/4
```

Configuring Maximum Idle Time for SSH Sessions

By default, SSH sessions do not time out. Optionally, you can set the amount of time an SSH session can be inactive before the Foundry device closes it. For example, to set the maximum idle time for SSH sessions to 30 minutes:

```
FastIron SuperX Switch(config)# ip ssh idle-time 30
```

Syntax: ip ssh idle-time <minutes>

If an established SSH session has no activity for the specified number of minutes, the Foundry device closes it. An idle time of 0 minutes (the default value) means that SSH sessions never time out. The maximum idle time for SSH sessions is 240 minutes.

Filtering SSH Access Using ACLs

You can permit or deny SSH access to the Foundry device using ACLs. To use ACLs, first create the ACLs you want to use. You can specify a numbered standard IPv4 ACL, a named standard IPv4 ACL.

Then enter the following command:

```
FastIron SuperX Switch(config)# access-list 10 permit host 192.168.144.241
FastIron SuperX Switch(config)# access-list 10 deny host 192.168.144.242 log
FastIron SuperX Switch(config)# access-list 10 permit host 192.168.144.243
FastIron SuperX Switch(config)# access-list 10 deny any
FastIron SuperX Switch(config)# ssh access-group 10
```

Syntax: ssh access-group <standard-named-acl> | <standard-numbered-acl>

Displaying SSH Connection Information

Up to five SSH connections can be active on the Foundry device. To display information about SSH connections, enter the following command:

```
FastIron SuperX Switch# show ip ssh
Connection Version  Encryption  Username   IP Address
1           SSH-2   3des-cbc    Hanuma     192.168.144.241
2           SSH-2   3des-cbc    Mikaila    192.168.144.241
3           SSH-2   3des-cbc    Jenny      192.168.144.241
4           SSH-2   3des-cbc    Mariah     192.168.144.241
5           SSH-2   3des-cbc    Logan      192.168.144.241
```

Syntax: show ip ssh [| begin <expression> | exclude <expression> | include <expression>]

This display shows the following information about the active SSH connections:

Table 11: SSH Connection Information

This Field...	Displays...
Connection	The SSH connection ID. This can be from 1 – 5.
Version	The SSH version number.
Encryption	The encryption method used for the connection.
Username	The user name for the connection.
IP Address	The client IP address.

The **show who** command also displays information about SSH connections. For example:

```
FastIron SuperX Switch#show who
Console connections:
established, monitor enabled, in config mode
2 minutes 17 seconds in idle
Telnet connections (inbound):
1 closed
2 closed
3 closed
4 closed
5 closed
Telnet connection (outbound):
6 closed
SSH connections:
1 established, client ip address 192.168.144.241, user is hanuma
1 minutes 16 seconds in idle
2 established, client ip address 192.168.144.241, user is Mikaila
you are connecting to this session
18 seconds in idle
3 established, client ip address 192.168.144.241, user is Jenny
1 minutes 39 seconds in idle
4 established, client ip address 192.168.144.242, user is Mariah
41 seconds in idle
5 established, client ip address 192.168.144.241, user is Logan
23 seconds in idle
```

Syntax: show who [| begin<expression> | exclude<expression> | include<expression>]

To terminate one of the active SSH connections, enter the following command:

```
FastIron SuperX Switch# kill ssh 1
```

Syntax: kill ssh <connection-id>

Using Secure Copy

Secure Copy (SCP) uses security built into SSH to transfer image and configuration files to and from the device. SCP automatically uses the authentication methods, encryption algorithm, and data compression level configured for SSH. For example, if password authentication is enabled for SSH, the user is prompted for a user name and password before SCP allows a file to be transferred. No additional configuration is required for SCP on top of SSH.

You can use SCP to copy files on the Foundry device, including the startup configuration and running configuration files, to or from an SCP-enabled remote host.

SCP is enabled by default and can be disabled. To disable SCP, enter the following command:

```
FastIron SuperX Switch(config)# ip ssh scp disable
```

Syntax: ip ssh scp disable | enable

NOTE: If you disable SSH, SCP is also disabled.

The following are examples of using SCP to transfer files from and to a Foundry device.

NOTE: When using SCP, you enter the **scp** commands on the SCP-enabled client, rather than the console on the Foundry device.

NOTE: Certain SCP client options, including -p and -r, are ignored by the SCP server on the Foundry device. If an option is ignored, the client is notified.

To copy a configuration file (c:\cfg\foundry.cfg) to the running configuration file on a Foundry device at 192.168.1.50 and log in as user terry, enter the following command on the SCP-enabled client:

```
C:\> scp c:\cfg\foundry.cfg terry@192.168.1.50:runConfig
```

If password authentication is enabled for SSH, the user is prompted for user terry's password before the file transfer takes place.

To copy the configuration file to the startup configuration file:

```
C:\> scp c:\cfg\foundry.cfg terry@192.168.1.50:startConfig
```

To copy the configuration file to a file called config1.cfg on the flash card in slot 1 on a management module:

```
C:\> scp c:\cfg\foundry.cfg terry@192.168.1.50:slot1:/config1.cfg
```

To copy the configuration file to a file called config1.cfg on the flash card in slot 2 on a management module:

```
C:\> scp c:\cfg\foundry.cfg terry@192.168.1.50:slot2:/config1.cfg
```

To copy the running configuration file on a Foundry device to a file called c:\cfg\fdryrun.cfg on the SCP-enabled client:

```
C:\> scp terry@192.168.1.50:runConfig c:\cfg\fdryrun.cfg
```

To copy the startup configuration file on a Foundry device to a file called c:\cfg\fdrystart.cfg on the SCP-enabled client:

```
C:\> scp terry@192.168.1.50:startConfig c:\cfg\fdrystart.cfg

To overwrite the running configuration file:

C:\> scp c:\cfg\foundry.cfg terry@192.168.1.50:runConfig-overwrite
```

Port Link Dampening

The port link dampening feature allows you to configure a wait period before a port, whose link goes down then up, becomes enabled.

If the port link state toggles (from down to up or from up to down) for a specified number of times within a specified period, the interface is physically disabled for the specified wait period. Once the wait period expires, the port's link state is re-enabled. However, if the wait period is set to zero (0) seconds, the port's link state will remain disabled until it is manually re-enabled.

Configuring Port Link Dampening on an Interface

This feature is configured at the interface level.

```
FastIron SuperX Switch(config)# interface ethernet 2/1
FastIron SuperX Switch(config-if-e10000-2/1)# link-error-disable 10 3 10
```

Syntax: [no] link-error-disable <toggle-threshold> <sampling-time-in-sec> <wait-time-in-sec>

The <toggle-threshold> is the number of times a port's link state goes from up to down and down to up before the wait period is activated. The default is 0. Enter a valid value range from 1-50.

The <sampling-time-in-sec> is the amount of time during which the specified toggle threshold can occur before the wait period is activated. The default is 0 seconds. Enter a value greater than 0 seconds.

The <wait-time-in-sec> is the amount of time the port remains disabled (down) before it becomes enabled. Entering 0 indicates that the port will stay down until an administrative override occurs. Enter a value greater than 0 seconds to configure a wait time.

Configuring Port Link Dampening on a Trunk

You can configure the port link dampening feature on the primary port of a trunk using the **link-error-disable** command. Once configured on the primary port, the feature is enabled on all ports that are members of the trunk. You cannot configure port link dampening on port members of the trunk.

Enter commands such as the following on the primary port of a trunk.

```
FastIron SuperX Switch(config)# interface ethernet 2/1
FastIron SuperX Switch(config-if-e10000-2/1)#link-error-disable 10 3 10
```

Re-enabling a Port Disabled by Port Link Dampening

A port disabled by port link dampening is automatically re-enabled once the wait period expires; however, if the wait period is set to zero (0) seconds, you must re-enable the port by entering the following command on the disabled port:

```
FastIron SuperX Switch(config)# interface ethernet 2/1
FastIron SuperX Switch(config-if-e10000-2/1)# no link-error-disable 10 3 10
```

Displaying Ports Configured with Port Link Dampening

Ports that have been disabled due to the port link dampening feature are identified in the output of the **show link-error-disable** command. The following shows an example output.

```
FastIron SuperX Switch(config)#show link-error-disable

Port 2/1 is forced down by link-error-disable.
```

Use the **show link-error-disable all** command to display the ports that have the port link dampening feature enabled. The following shows an example output.

```
FastIron SuperX Switch(config)#show link-error-disable all

Port8/1 is configured for link-error-disable
            threshold:1, sampling_period:10, waiting_period:0
Port8/2 is configured for link-error-disable
            threshold:1, sampling_period:10, waiting_period:0
Port8/3 is configured for link-error-disable
            threshold:1, sampling_period:10, waiting_period:0
Port8/4 is configured for link-error-disable
            threshold:1, sampling_period:10, waiting_period:0
Port8/5 is configured for link-error-disable
            threshold:4, sampling_period:10, waiting_period:2
Port8/9 is configured for link-error-disable
             threshold:2, sampling_period:20, waiting_period:0
```

Syntax: show link-error-disable [all]

DNS List

This section describes the Domain Name Server (DNS) resolver feature.

Configuring Domain Name List and Domain Look Up

The DNS resolver is a feature in a Layer 2 Switch or Layer 3 Switch that sends and receives queries to and from the DNS server on behalf of a client. Prior to this release, the feature lets you use one domain name to perform Telnet, ping, traceroute and other DNS query commands. You define one domain name on a Foundry Layer 2 Switch or Layer 3 Switch and up to four DNS servers. Host names and their IP addresses are configured on the DNS servers.

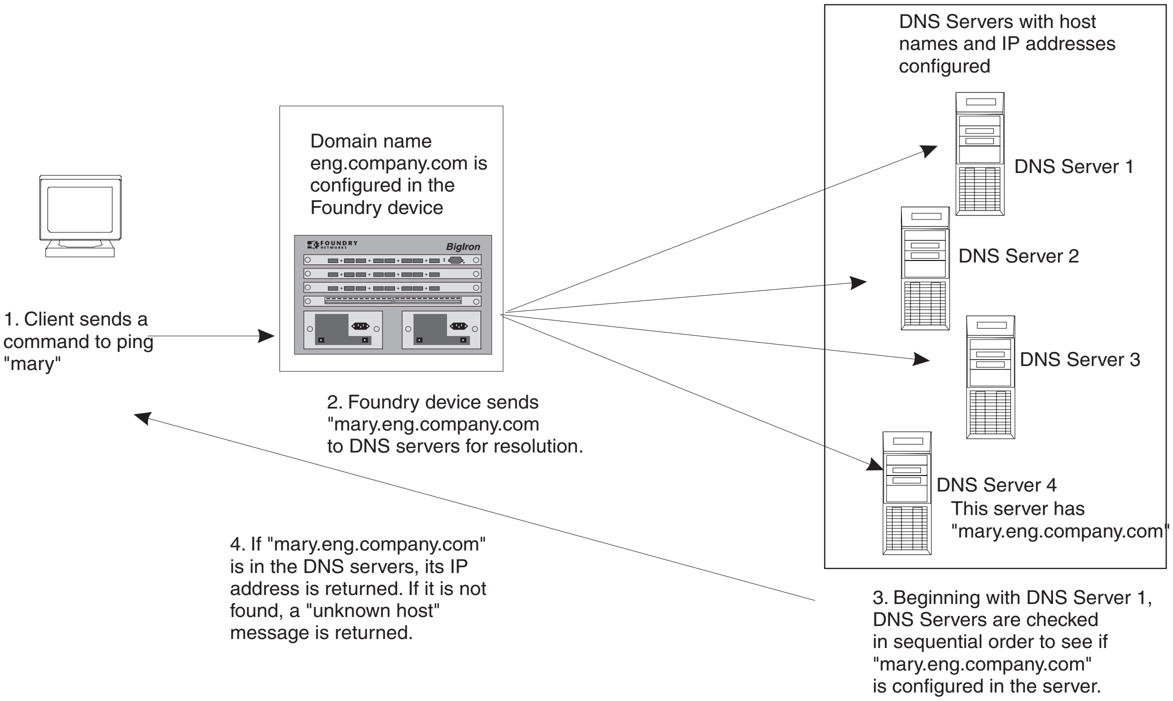
When a client performs a DNS query, all hosts within that domain can be recognized. After you define a domain name, the Foundry Layer 2 Switch or Layer 3 Switch automatically appends the appropriate domain to a host and forwards it to the DNS servers for resolution.

For example, if the domain “ds.company.com” is defined on a Foundry Layer 2 Switch or Layer 3 Switch and you want to initiate a ping to “mary”. You need to reference only the host name instead of the host name and its domain name. For example, you could enter the following command to initiate the ping:

```
U:>ping mary
```

The Foundry Layer 2 Switch or Layer 3 Switch qualifies the host name by appending a domain name. For example, `mary.ds1.company.com`. This qualified name is sent to the DNS server for resolution. If there are four DNS servers configured, it is sent to the first DNS server. If the host name is not resolved, it is sent to the second DNS server. If a match is found, a response is sent back to the client with the host’s IP address. If no match is found, a “unknown host” message is returned. (See Figure 12.)

Figure 12 DNS resolution with one domain name


DNS Servers with host names and IP addresses configured
Domain name eng.company.com is configured in the Foundry device
FOUNDRY NETWORKS
BigIron
DNS Server 1
DNS Server 2
DNS Server 3
DNS Server 4
This server has "mary.eng.company.com"
1. Client sends a command to ping "mary"
2. Foundry device sends "mary.eng.company.com to DNS servers for resolution.
4. If "mary.eng.company.com" is in the DNS servers, its IP address is returned. If it is not found, a "unknown host" message is returned.
3. Beginning with DNS Server 1, DNS Servers are checked in sequential order to see if "mary.eng.company.com" is configured in the server.

Beginning with software release 03.0.00, you can create a list of domain names that can be used to resolve host names. This list can have more than one domain name. When a client performs a DNS query all hosts within the domains in the list can be recognized and queries can be sent to any domain on the list.

Defining a Domain Name

If you want to define only one domain to resolve host names, enter a command such as the following:

```
FastIron SuperX Switch(config)# ip dns domain-name ds.company.com
```

Syntax: [no] ip dns domain-name <domain-name>

Enter the domain name for <domain-name>.

Defining a Domain List

If you want to use more than one domain name to resolve host names, you can create a list of domain names. For example, enter the commands such as the following:

```
FastIron SuperX Switch(config)# ip dns domain-list company.com
FastIron SuperX Switch(config)# ip dns domain-list ds.company.com
FastIron SuperX Switch(config)# ip dns domain-list hw_company.com
FastIron SuperX Switch(config)# ip dns domain-list qa_company.com
FastIron SuperX Switch(config)#
```

The domain names are tried in the order you enter them

Syntax: [no] ip dns domain-list <domain-name>

Defining DNS Servers

You can configure the Foundry device to recognize up to four DNS servers. The first entry serves as the primary default address. If a query to the primary address fails to be resolved after three attempts, the next DNS address is queried (also up to three times). This process continues for each defined DNS address until the query is resolved. The order in which the default DNS addresses are polled is the same as the order in which you enter them.

To define DNS servers, enter a command such as the following:

```
FastIron SuperX Switch(config)# ip dns server-address 209.157.22.199 205.96.7.15
208.95.7.25 201.98.7.15
```

Syntax: [no] ip dns server-address <ip-addr> [<ip-addr>] [<ip-addr>] [<ip-addr>]

In this example, the first IP address entered becomes the primary DNS address and all others are secondary addresses. Because IP address 201.98.7.15 is the last address listed, it is also the last address consulted to resolve a query.

Using a DNS Name To Initiate a Trace Route

Suppose you want to trace the route from a Foundry Layer 3 Switch to a remote server identified as NYC02 on domain newyork.com. Because the NYC02@ds1.newyork.com domain is already defined on the Layer 3 Switch, you need to enter only the host name, NYC02, as noted below.

```
FastIron SuperX Switch# traceroute nyc02
```

Syntax: traceroute <host-ip-addr> [maxttl <value>] [minttl <value>] [numeric] [timeout <value>] [source-ip <ip addr>]

The only required parameter is the IP address of the host at the other end of the route. See the *Foundry Switch and Router Command Line Interface Reference* for information about the parameters.

After you enter the command, a message indicating that the DNS query is in process and the current DNS address (IP address of the domain name server) being queried appear on the screen:

```
Type Control-c to abort
Sending DNS Query to 209.157.22.199
Tracing Route to IP node 209.157.22.80
To ABORT Trace Route, Please use stop-traceroute command.
 Traced route to target IP node 209.157.22.80:
   IP Address          Round Trip Time1    Round Trip Time2
 207.95.6.30         93 msec             121 msec
```

In the above example, 209.157.22.199 is the IP address of the domain name server (default DNS address), and 209.157.22.80 represents the IP address of the NYC02 host.

Enhancements to Username and Password

This section describes the enhancements to the username and password features in release 03.0.00.

The following rules are enabled by default:

- Users are now required to accept the message of the day.
- Users are locked out (disabled) if they fail to login after three attempts. This feature is automatically enabled in release 03.0.00. You can use the **disable-on-login-failure** command to change the number of login attempts (up to 10) before users are locked out.

The following rules are disabled by default:

- Enhanced user password combination requirements
- User password masking
- Quarterly updates of user passwords
- You can configure the system to store up to 15 previously configured passwords for each user.
- You can use the **disable-on-login-failure** command to change the number of login attempts (up to 10) before users are locked out.
- A password can now be set to expire.

Enabling Enhanced User Password Combination Requirements

When strict password enforcement is enabled on the Foundry device, you must enter a minimum of eight characters containing the following combinations when you create an enable and a user password:

- At least two upper case characters
- At least two lower case characters
- At least two numeric characters
- At least two special characters

NOTE: Password minimum and combination requirements are strictly enforced.

Use the **enable strict-password-enforcement** command to enable the password security feature.

```
FastIron SuperX Switch(config)# enable strict-password-enforcement
```

Syntax: [no] enable strict-password-enforcement

This feature is disabled by default.

NOTE: When you upgrade to release 03.0.00 or later, the old passwords are still valid; however, users must change their passwords to follow the new format to take advantage of this password enhancement.

Enabling User Password Masking

By default, when you use the CLI to create a user password, the password displays on the console as you type it. For enhanced security, you can configure the Foundry device to mask the password characters entered at the CLI. When password masking is enabled, the CLI displays asterisks (*) on the console instead of the actual password characters entered.

The following shows the default CLI behavior when configuring a username and password:

```
FastIron SuperX Switch(config)# username kelly password summertime
```

The following shows the CLI behavior when configuring a username and password when **password-masking** is enabled:

```
FastIron SuperX Switch(config)# username kelly password
Enter Password: ********
```

NOTE: When password masking is enabled, press the [Enter] key before entering the password.

Syntax: username <name> password [Enter]

For [Enter], press the Enter key. Enter the password when prompted.

If **strict-password-enforcement** is enabled, enter a password which contains the required character combination. See "Enabling Enhanced User Password Combination Requirements" on page 74.

To enable password masking, enter the following command:

```
FastIron SuperX Switch(config)# enable user password-masking
```

Syntax: [no] enable user password-masking

Enabling User Password Aging

For enhanced security, password aging enforces quarterly updates of all user passwords. After 180 days, the CLI will automatically prompt users to change their passwords when they attempt to sign on.

Password aging is disabled by default. To enable it, enter the following command at the global CONFIG level of the CLI:

```
FastIron SuperX Switch(config)# enable user password-aging
```

Syntax: [no] enable user password-aging

Enabling Enhanced Password History

By default, the Foundry device stores the last five user passwords for each user. When changing a user password, the user cannot use any of the five previously configured passwords.

For security purposes, you can configure the Foundry device to store up to 15 passwords for each user, so that users do not use the same password multiple times. If a user attempts to use a password that is stored, the system will prompt the user to choose a different password.

To configure enhanced password history, enter a command such as the following at the global CONFIG level of the CLI:

```
FastIron SuperX Switch(config)# enable user password-history 15
```

Syntax: [no] enable user password-history <1 – 15>

Requirement to Accept the Message of the Day

If a message of the day (MOTD) is configured, a user will be required to press the Enter key before he or she can login. MOTD is configured using the **banner motd** command.

There are no new CLI commands for this feature.

Enhanced Login Lockout

The CLI has been enhanced to provide up to three login attempts. If a user fails to login after three attempts, that user is locked out (disabled). If desired, you can increase or decrease the number of login attempts before the user is disabled. To do so, enter a command such as the following at the global CONFIG level of the CLI:

```
FastIron SuperX Switch(config)# enable user disable-on-login-failure 7
```

Syntax: enable user disable-on-login-failure <1 – 10>

To re-enable a user that has been locked out, do one of the following:

- Reboot the Foundry device to re-enable all disabled users.
- Enable the user by entering the following command:

  ```
  FastIron SuperX Switch(config)# username sandy enable
  ```

For example:

```
FastIron SuperX Switch(config)# user sandy enable
FastIron SuperX Switch(config)# show user
Username  Password                          Encrypt   Priv Status   Expire Time
==============================================================================

sandy     $1$Gz...uX/$wQ44fVGtsqbKWkQknzAZ6. enabled   0   enabled  90 days
```

Syntax: username <name> enable

Setting Passwords to Expire

You can set a user password to expire. Once a password expires, the administrator must assign a new password to the user. To configure a user password to expire, enter the following:

```
FastIron SuperX Switch(config)# username sandy expires 20
```

Syntax: username <name> expires <days>

Enter 1 – 365 for number of days. The default is 90 days.

For example:

```
FastIron SuperX Switch(config)# username sandy expires 20
FastIron SuperX Switch(config)# show user
Username     Password                           Encrypt   Priv  Status   Expire Time
==================================================================================
sandy       $1$Gz...uX/$wQ44fVGtsqbKWkQknzAZ6. enabled    0    enabled  20 days
```

Configuring IPG

You can configure the Interpacket Gap (IPG), which is the time delay, in bit time, between frames transmitted by the device. You configure IPG at the interface level. The command you use depends on the interface type on which IPG is being configured.

The default interpacket gap is 96 bits-time, which is 9.6 microseconds for 10 Mbps Ethernet, 960 nanoseconds for 100 Mbps Ethernet, 96 nanoseconds for 1 Gbps Ethernet, and 9.6 nanoseconds for 10 Gbps Ethernet.

Configuration Notes

When configuring IPG, note the following:

- IPG configuration commands are based on "port regions". All ports within the same port region should have the same IPG configuration. If a port region contains two or more ports, changes to the IPG configuration for one port are applied to all ports in the same port region. When you enter a value for IPG, the CLI displays the ports to which the IPG configuration is applied. For example:

  ```
  FESX424 Router(config-if-e1000-7/1)#ipg-gmii 120
  IPG 120(112) has been successfully configured for ports 7/1 to 7/12
  ```

- When you enter a value for IPG, the device applies the closest valid IPG value for the port mode to the interface. For example, if you specify 120 for a 1 Gigabit Ethernet port in 1 Gigabit mode, the device assigns 112 as the closest valid IPG value to program into hardware.

Configuring IPG on a Gigabit Ethernet Port

On a Gigabit Ethernet port, you can configure IPG for 10/100 mode and for Gigabit Ethernet mode.

10/100M mode

To configure IPG on a Gigabit Ethernet port for 10/100M mode, enter the following command.

```
FESX424 Router(config)# interface ethernet 7/1
FESX424 Router(config-if-e1000-7/1)# ipg-mii 120
IPG 120(120) has been successfully configured for ports 7/1 to 7/12
```

Syntax: [no] ipg-mii <bit time>

Enter 12-124 for <bit time>. The default is 96 bit time.

1G Mode

To configure IPG on a Gigabit Ethernet port for 1-Gigabit Ethernet mode, enter commands such as the following:

```
FESX424 Router(config)# interface ethernet 7/1
FESX424 Router(config-if-e1000-7/1)# ipg-gmii 120
IPG 120(112) has been successfully configured for ports 7/1 to 7/12
```

Syntax: [no] ipg-gmii <bit time>

Enter 48 - 112 for <bit time>. The default is 96 bit time.

Configuring IPG on a 10-Gigabit Ethernet Interface

To configure IPG on a 10-Gigabit Ethernet interface, enter commands such as the following:

```
FESX424 Router(config)# interface ethernet 9/1
FESX424 Router(config-if-e10000-9/1)# ipg-xgmii 120
IPG 120(128) has been successfully configured for port 9/1
```

Syntax: [no] ipg-xgmii <bit time>

Enter 96-192 for <bit time>. The default is 96 bit time.

Displaying VLANs in Alphanumeric Order

In releases prior to 03.0.00, the output of some **show** commands list VLANs in the order that they were configured. Starting with release 03.0.00, by default, the VLANs are displayed in alphanumeric order.

For example, in releases prior to 03.0.00, if you configure VLANs in the order VLAN 10, VLAN 100, then VLAN 2, the **show run** command output displays:

```
FastIron SuperX Switch(config)# show run
...

vlan 10 by port
...
vlan 100 by port
...
vlan 2 by port
...
```

Starting with release 03.0.00, VLANs are displayed in alphanumeric order, as shown in the following example.

```
FastIron SuperX Switch(config)# show run
...

vlan 2 by port
...
vlan 10 by port
...
vlan 100 by port
...
```

Disabling Support for POE Legacy Power Consuming Devices

Foundry's POE devices automatically support most legacy power consuming devices (non-802.3af compliant devices), as well as all 802.3af-compliant devices. In releases prior to 03.0.00, support for legacy devices is always enabled on Foundry's POE devices and cannot be disabled. Starting with release 03.0.00, if desired, you can disable and re-enable support for legacy POE power consuming devices on a global basis (on the entire device) or on individual slots (chassis devices only). When you disable legacy support, 802.3af-compliant devices are not affected.

To disable support for legacy power consuming devices on a global basis, enter the following command at the Global CONFIG level of the CLI:

```
FastIron SuperX Switch(config)# no legacy-inline-power
```

On chassis devices (FSX, FSX 800, and FSX 1600), you can disable support for legacy power consuming devices per slot. To disable support on a slot, enter a command such as the following at the Global CONFIG level of the CLI:

```
FastIron SuperX Switch(config)# no legacy-inline-power 2
```

The above command disables legacy support on all ports in slot 2.

Syntax: [no] legacy-inline-power [<slotnum>/]

To re-enable support for legacy power consuming devices after it has been disabled, enter the **legacy-inline-power** command (without the **no** parameter).

<slotnum> is required on chassis devices when disabling or re-enabling legacy support on a slot.

Use the **show run** command to view whether support for POE legacy power consuming devices is enabled or disabled.

SNMP Version 3 Traps

Starting with release 03.0.00, an SNMP agent upgrade supports SNMP notifications in SMIv2 format. This allows notifications to be encrypted and sent to the target hosts in a secure manner.

Defining an SNMP Group and Specifying Which View is Notified of Traps

In software release 03.0.00, the SNMP group command allows configuration of a viewname for notification purpose, similar to the read and write view. The default viewname is "all", which allows access to the entire MIB.

To configure an SNMP user group, first configure SNMP v3 views using the **snmp-server view** command. (See the *Foundry Security Guide* for information on configuring views). Then enter a command such as the following:

```
FastIron SuperX Switch(config)# snmp-server group admin v3 auth read all write all
notify all
```

Syntax: [no] snmp-server group <groupname>
v1 | v2 | v3
auth | noauth | priv
[access <standard-acl-id>] [read <viewstring> | write <viewstring> | notify <viewstring>]

The **group** <groupname> parameter defines the name of the SNMP group to be created.

The **v1**, **v2**, or **v3** parameter indicates which version of SNMP to use. In most cases, you will use v3, since groups are automatically created in SNMP versions 1 and 2 from community strings.

The **auth** | **noauth** parameter determines whether or not authentication will be required to access the supported views. If auth is selected, then only authenticated packets are allowed to access the view specified for the user group. Selecting **noauth** means that no authentication is required to access the specified view. Selecting **priv** means that an authentication password will be required from the users.

The **access** <standard-acl-id> parameter is optional. It allows incoming SNMP packets to be filtered based on the standard ACL attached to the group.

The **read** <viewstring> | **write** <viewstring> parameter is optional. It indicates that users who belong to this group have either read or write access to the MIB.

The **notify** view allows administrators to restrict the scope of varbind objects that will be part of the notification. All of the varbinds need to be in the included view for the notification to be created.

The <viewstring> variable is the name of the view to which the SNMP group members have access. If no view is specified, then the group has no access to the MIB.

Defining the UDP Port for SNMP v3 Traps

Starting in release 03.0.00, SNMP host command enhancements allow configuration of notifications in SMIv2 format, with or without encryption, in addition to the previously supported SMIv1 trap format.

You can define a port that receives the SNMP v3 traps by entering a command such as the following:

```
FastIron SuperX Switch(config)# snmp-server host 192.168.4.11 version v3 auth
security-name port 165
```

Syntax: [no] snmp-server host <ip-address> version [v1 | v2c <community-string> | v3 auth | noauth | priv <security-name>] [port <trap-UDP-port-number>]

The <ip-address> parameter specifies the IP address of the host that will receive the trap.

For **version**, indicate one of the following:

For SNMP version 1, enter **v1** and the name of the community string (<community-string>). This string is encrypted within the system.

NOTE: The options "v2c" and "v3" are new in software release 03.0.00. If the configured version is v2c, then the notification is sent out in SMIv2 format, using the community string, but in cleartext mode. To send the SMIv2 notification in SNMPv3 packet format, configure v3 with auth and/or privacy parameters by specifying a security name. The actual authorization and privacy values are obtained from the security name.

For SNMP version 2c, enter **v2** and the name of the community string. This string is encrypted within the system.

For SNMP version 3, enter one of the following depending on the authorization required for the host:

- **v3 auth** <security-name>: Allow only authenticated packets.
- **v3 no auth** <security-name>: Allow all packets.
- **v3 priv** <security-name>: A password is required

For **port** <trap-UDP-port-number>, specify the number of the UDP port number on the host that will receive the trap.

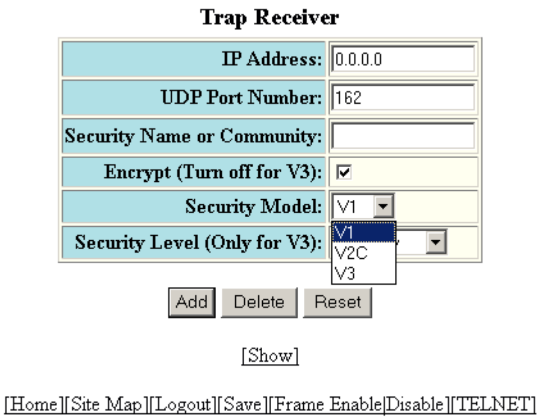
Using the Web Interface

The Web management page for “Trap Receiver” is used to configure and list all the trap hosts for this device. In release 03.0.00, this feature was updated to support the version, security model and the security level.

Selecting the Security Model

When configuring the security model, the **V1**, **V2C**, or **V3** parameter indicates which version of SNMP to use. In most cases, you will be using v3, since groups are automatically created in SNMP versions 1 and 2 from community strings.

Select the version of SNMP to be used from the Security Model drop down box.



Trap Receiver

IP Address:	0.0.0.0
UDP Port Number:	162
Security Name or Community:	
Encrypt (Turn off for V3):	☑
Security Model:	V1
Security Level (Only for V3):	

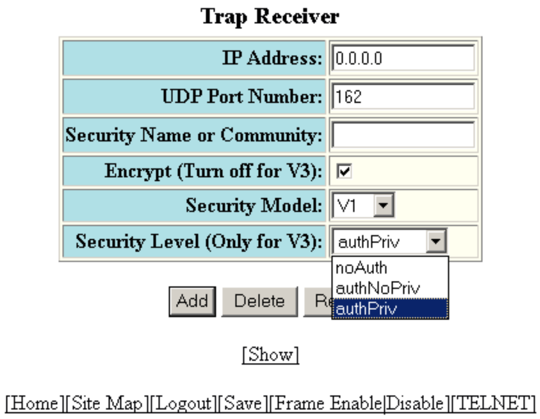
V1
V2C
V3
Add
Delete
Reset
[Show]
[Home][Site Map][Logout][Save][Frame Enable|Disable][TELNET]

Selecting the Security Level

For SNMP version 3, select one of the following security levels depending on the authorization required for the host:

- **No auth** - All packets are allowed.
- **auth No Priv** - Only authenticated packets are allowed.
- **auth Priv** - A password is required

Figure 13 New Trap Receiver Security Level



Trap Receiver

IP Address:	0.0.0.0
UDP Port Number:	162
Security Name or Community:	
Encrypt (Turn off for V3):	☑
Security Model:	V1
Security Level (Only for V3):	authPriv

noAuth
authNoPriv
authPriv
Add
Delete
Re
[Show]
[Home][Site Map][Logout][Save][Frame Enable|Disable][TELNET]

Trap MIB Changes

In release 03.0.00, the Foundry Enterprise Trap MIB was rewritten in SMIv2 format, as follows:

- The MIB name was changed from FOUNDRY-SN-TRAP-MIB to FOUNDRY-SN-NOTIFICATION-MIB
- Individual notifications were changed to NOTIFICATION-TYPE instead of TRAP-TYPE.
- As per the SMIv2 format, each notification now has an OID associated with it. The root node of the notification is snTraps (OID: enterprise.foundry.0). For example, OID for snTrapRunningConfigChanged is {snTraps.73}. Earlier, each trap had a trap ID associated with it, as per the SMIv1 format.

Backward compatibility with SMIv1 trap format

The Foundry device will continue to support creation of traps in SMIv1 format, as before. To allow the device to send notifications in SMIv2 format, configure the device as described above. The default mode is still the original SMIv1 format.

SNMP MIB Changes

In release 03.0.00, the following changes were made to the SNMP MIBs:

- Enterprise Trap MIB rewritten in SNMPv2 format
- Support for RFC 3411 – SNMP Framework MIB
- Support for RFC 3412 - Message Processing and Dispatching (MPD) for the SNMP MIB.
- Support for RFC 3413 – SNMP Target MIB
- Support for RFC 3414 – User-Security Model for SNMPv3 MIB
- Support for RFC 3415 – View-based Access Control Model for SNMP MIB
- Changes to snAgTrpRcvrTable:
- The Object snAgTrpRcvrCommunity has been renamed to snAgTrpRcvrCommunityOrSecurityName to indicate that this object can also have the v3 security name (user name), instead of community string.

New MIB Objects

Name, OID, and Syntax	Access	Description
snAgTrpRcvrSecurityModel fdry.1.3.6.1.4.1.1991.1.1.2.3.1.1.6 Syntax: INTEGER	read-write	Allows configuration of security model (v1, v2c or 3).
snAgTrpRcvrSecurityLevel fdry.1.3.6.1.4.1.1991.1.1.2.3.1.17 Syntax: INTEGER	read-write	Allows configuration of the security level (noauth, auth or auth+priv).

Base Layer 3

For information on how to configure static IP and RIP in the Base Layer 3 software image see the "Configuring Base Layer 3" chapter of the *Foundry FastIron X-Series Configuration Guide*.

For information about the other IP configuration commands in the Layer 2 with Base Layer 3 image, see the "Configuring IP" chapter of the *Foundry FastIron X-Series Configuration Guide*.

Full Layer 3

For information about full Layer 3 protocols and how to configure them, see the *Foundry FastIron X-Series Configuration Guide.*

Where To Get More Information

These release notes provide basic setup information. For more advanced configuration information, see the documents listed in Table 12.

Table 12: Feature Documentation

Title	Contents
Foundry FastIron X-Series Configuration Guide	Provides configuration procedures for system-level features and enterprise routing protocols such as IP, RIP, IP multicast, OSPF, BGP4, VRRP and VRRP-E.
Foundry FastIron Stackable Hardware Installation Guide	For FES, FESX and FWSX devices, provides the following information: • Product Overview • Installation instructions • Hardware Specifications
Foundry FastIron X-Series Chassis Hardware Installation Guide	For FSX, FSX 800, and FSX 800 devices, provides the following information: • Product Overview • Installation instructions • Hardware Specifications
Foundry Security Guide	Procedures for securing management access to Foundry devices and for protecting against Denial of Service (DoS) attacks.
Foundry *Management Information Base Reference*	Simple Network Management Protocol (SNMP) Management Information Base (MIB) objects.

Software Fixes

This section lists the software issues that were fixed in releases 03.0.01 through 03.0.01c. The **P** column indicates the priority of the software fix, as follows:

- 0 = Critical
- 1 = Major
- 2 = Medium
- 3 = Minor

The software fixes are sorted by category, then by priority.

The following table lists the software issues that were fixed in release 03.0.01c.

Table 13: Software Fixes in Release 03.0.01c

Category	P	Description	Bug ID #
System	1	Symptom: Adding or deleting a port from a VLAN may result in high CPU utilization, if the port is already tagged into a large VLAN range. This may result in the control protocol timing out, or a system reset. Resolution: Fixed in release 03.0.01c	63797

The following table lists the software issues that were fixed in release 03.0.01b.

Table 14: Software Fixes in Release 03.0.01b

Category	P	Description	Bug ID #
Other	1	Symptom: The fan status is not properly displayed in the output of the **show chassis** command. Resolution: Fixed in release 03.0.01b	62522

The following table lists the software issues that were fixed in software release 03.0.01a

Table 1: Software Fixes in Release 03.0.01a

Category	P	Description	Bug ID #
AAA	2	Symptom: When AAA authentication is in effect, the Foundry device does not accept a username entry at the console login prompt. Instead, the Foundry device displays a "User Login Failure" message then automatically enters a username, causing an invalid login. Resolution: Fixed in 03.0.01a	62977
AAA,CLI	2	Symptom: The following problems occur when using the console port to log into the Foundry device: • When AAA and TACACs authentication is in effect, the Foundry device does not accept a username entry at the console login prompt and instead returns a "User Login Failure" message. • When AAA and local authentication is used, the Foundry device does not encrypt the username password entered at the console login prompt. Resolution: Fixed in 03.0.01a	63016
CLI	2	Symptom: The Foundry device does not show the MOTD banner during console login. Resolution: Fixed in 03.0.01a	62978
IP Stack	2	Symptom: The Foundry device will not learn new MAC addresses on an interface if the Interface configuration is removed then re-added. Resolution: Fixed in 03.0.01a	62115

Table 1: Software Fixes in Release 03.0.01a

Category	P	Description	Bug ID #
IP Stack	2	Symptom: The CLI command **clear mac** does not clear router interfaces from the MAC table. Resolution: Fixed in 03.0.01a	62116
IPv4 Forwarding	2	Symptom: The Foundry device randomly loses Layer 2 connectivity to directly attached devices. Resolution: Fixed in 03.0.01a	62412
OSPF	1	Symptom: The OSPF distribute-list filter does not correctly filter routes. This can be seen in the **show ip route** command output. Resolution: Fixed in 03.0.01a	58305, 62556, and 62557
Other	1	Symptom: The Foundry device locks up after the **dm diag** command is issued, and while the device is reloading the software. Resolution: Fixed in 03.0.01a	62393
SNMP	1	Symptom: The IronView Network Manager shows that some traps are unsupported. Resolution: Fixed in 03.0.01a	60550
SYSLOG, System	2	Symptom: In a configuration with SNTP, the **SysUpTime** probe incorrectly goes forward approximately six hours in an 80-hour time period. Resolution: Fixed in 03.0.01a	62019, 62665
System	1	Symptom: The power supply fan speed default setting is not correct. Resolution: Fixed in release 03.0.01a	63196
Tagging	2	The Foundry device may reload the software when the **no dual-mode** command is applied. Resolution: Fixed in 03.0.01a	63061

The following table lists the software issues that were fixed in release 03.0.01.

Table 15: Software Fixes in Release 03.0.01

Category	P	Description	Bug ID #
Access Lists	1	Symptom: The Foundry device does not remove an ACL from the configuration after the ACL binding fails. In addition, the ACL cannot be manually removed from the configuration. Resolution: Fixed in 03.0.01	57524
Auto-Negotiation	2	Symptom: After manually setting auto-negotiation to MDI, the output of the **show interface** command shows that auto-negotiation is set to MDIX. Resolution: Fixed in 03.0.01	60666
IGMP	1	Symptom: In certain IGMP configurations, a software reload occurs when active IGMP mode is enabled (CLI command **ip multicast active**). Resolution: Fixed in 03.0.01	61268

Table 15: Software Fixes in Release 03.0.01 (Continued)

Category	P	Description	Bug ID #
IGMP	2	Symptom: The following error message appears on the console: `WARNING: Looping packet detected on port 9/12 with source IP 0.0.0.0.` Resolution: Fixed in 03.0.01	61321
IP Stack	1	Symptom: A port on the Foundry device learns an incorrect supernet route via OSPF when the port flaps (fluctuates between up and down states). Resolution: Fixed in 03.0.01	62171
LACP	2	Symptom: The Foundry device does not clear a four-port LACP trunk group after the LACP configuration is removed. Resolution: Fixed in 03.0.01	60767
Other	1	Symptom: The Foundry device is unable to establish connectivity to a PC when connected to the switch in boot mode. Resolution: Fixed in 03.0.01	58140
SNMP Management	1	Symptom: The SNMP MIB for Chassis serial number, snChasSerNum, does not return a value. Resolution: Fixed in 03.0.01	57080
SNMP Management	1	Symptom: "Out of memory" error messages occur while attempting to create RMON history. Resolution: Fixed in 03.0.01. See also “Specifying the Maximum Number of Entries Allowed in the RMON Control Table” on page 23.	61886
Traffic policy	1	Symptom: The Foundry device does not properly bind or unbind an ACL from a virtual interface if the ACL includes a traffic policy as well as the keyword "established" in the same filter definition. When binding or unbinding an ACL under these circumstances, the system will return an error message. Resolution: Fixed in 03.0.01	61388
VSRP	3	Symptom: The **show vsrp** command output displays an incorrect value in the "priority" field. **NOTE:** This is a display issue only and does not affect VSRP functionality. Resolution: Fixed in 03.0.01	60671
Web Management	1	Symptom: A software reload occurs when the Foundry device fails to extract a line from a large buffer. Resolution: Fixed in 03.0.01	43428