JUNIPER NETWORKS

Junos® OS

Designing and Implementing a Junos Node Unifier Network

Release
1.0

Published: 2012-08-13

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Junos® OS Designing and Implementing Junos Node Unifier
Release 1.0

Revision History
August 2012—R1 Junos Node Unifier 1.0

The information in this document is current as of the date on the title page.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at http://www.juniper.net/support/eula.html. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

Part 1 Introduction to Junos Node Unifier

Chapter 1 **Introduction to Junos Node Unifier . . . 3**

Audience . . . 3
Junos Node Unifier Overview . . . 3
Basic Architecture of a JNU Network . . . 4
Terms Used in the JNU Documentation . . . 4
Understanding the JNU Software Architecture . . . 5
Management Plane . . . 5
Data Plane . . . 5

Part 2 Planning a JNU Implementation

Chapter 2 **Planning Overview . . . 9**

Platform Considerations for the JNU Controller . . . 9
Platform Considerations for JNU Satellite Devices . . . 9

Part 3 Implementing JNU

Chapter 3 **Getting Started with the JNU Software . . . 13**

Installing the JNU Software on the Controller . . . 13
Installing the JNU Software on the Satellite Devices . . . 13
Initializing JNU Mode on the Controller . . . 14
Initializing JNU Mode on the Satellite Devices . . . 21

Chapter 4 **Using SNMP and System Logging with JNU . . . 27**

Centralized Collection of SNMP Statistics and Log Messages . . . 27
Collecting Log Messages . . . 27
Overview of Using the JNU Controller as an SNMP Proxy Agent . . . 27
Configuring the JNU Controller as an SNMP Proxy Agent . . . 28

Chapter 5 **Configuring Junos OS Features with JNU . . . 31**

Configuring Junos Features with JNU Configuration Templates . . . 31
Displaying a List of Configuration Templates . . . 31
Displaying the Configuration Parameters in a Template . . . 31
Configuring the Template . . . 32
Committing the Configuration . . . 32
Configuring Junos Features with JNU Free Form . . . 32

Chapter 6 **Committing Configurations . . . 35**

Commit Process for Satellites Already Connected to the Controller . . . 35
Commit Process for Satellite Devices That Come Online After the Commit Process on the Controller . . . 36

Chapter 7 JNU Operational Mode Commands 37
jnu-add-delete-satellites 38
config-free-form 39
jnu-initialize-controller 40
jnu-commit 44
jnu-order-satellites 45
op 46
jnu-remote 47
jnu-rollback 49
jnu-show-satellites 50
jnu-show-configuration 51
Chapter 8 Setting Up a Basic JNU Implementation 55
Example: Setting Up a Basic JNU Implementation 55

PART 1

Introduction to Junos Node Unifier

- Introduction to Junos Node Unifier on page 3

CHAPTER 1

Introduction to Junos Node Unifier

- Audience on page 3
- Junos Node Unifier Overview on page 3
- Basic Architecture of a JNU Network on page 4
- Terms Used in the JNU Documentation on page 4
- Understanding the JNU Software Architecture on page 5

Audience

This guide is intended to assist service providers to design and plan an implementation for Junos Node Unifier (JNU). We intend the guide to be used by the following:

- Network architects—Responsible for creating the overall design and architecture of the dual-stack network.
- Network planners—Responsible for planning the implementation from a network perspective, including equipment.
- Network operations engineer—Responsible for creating the configuration that implements the overall design. Also responsible for deploying the implementation and actively monitoring the network.
- Sales engineers—Responsible for working with architects, planners, and operations engineers to design and implement the network solution.

Junos Node Unifier Overview

Junos Node Unifier (JNU) allows you to configure and manage many Juniper Networks platforms from one MX Series router. You can use JNU to manage thousands of 1- and 10-Gigabit Ethernet ports in a single site or that are distributed across multiple sites from a single point.

JNU provides single-touch provisioning from one MX Series router acting as a controller. It provides a single point of:

- Configuration and management
- Running operational mode commands

- SNMP polling and SNMP traps
- Upgrading and downgrading platforms
- Collecting logging information

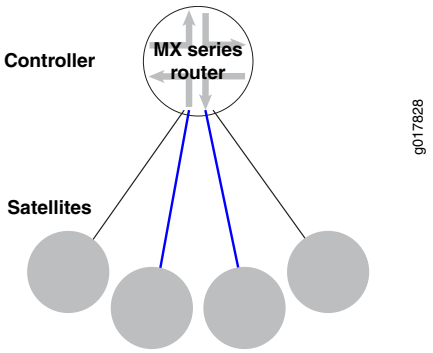
The JNU software answers the following needs:

- Ethernet port fanout or port multiplexer to control thousands of Ethernet ports from one MX Series router.
- Layer 2 switching on managed devices to meet Data Center needs, such as server port aggregation.
- Layer 3 MPLS routing on managed devices to provide business access and mobile backhaul applications.

Basic Architecture of a JNU Network

The basic architecture of a JNU implementation is a star configuration with one MX Series router acting as a hub to the connected satellite devices. The satellite devices are devices running the Juniper Networks Junos operating system (Junos OS), such as EX Series Ethernet switches and QFX Series devices.

Figure 1: Basic JNU Architecture



Terms Used in the JNU Documentation

Table 1 on page 4 defines terms used in the JNU documentation.

Table 1: JNU Terms

Term	Definition
Controller	An MX Series router that is used to manage and configure satellite devices.
JNU	Junos Node Unifier.
Satellite	Juniper Networks platforms that are managed by the controller.

Understanding the JNU Software Architecture

The JNU software architecture provides a separate management and data plane. This design provides maximum performance and reliability and the ability to efficiently scale the JNU network.

Management Plane

The JNU software uses a private management plane on the MX Series controller to manage satellite devices as follows:

- Provision satellite devices
- Operate satellite devices
- Perform SNMP polling and trap collection
- Collect logs
- Upgrade and downgrade software

The management connection between the controller and each satellite device runs on the management plane. The connection runs on a private IP subnet that is separated by dedicated virtual LAN tags so that it is separate from data traffic. In addition Network Address Translation (NAT) is used to translate the source address of traffic sent to the SNMP server so that all SNMP traffic from the satellite devices originates from a source address on the controller.

Data Plane

The data plane of the controller and satellite devices, which is responsible for forwarding user data and service traffic, is separate from the management plane.

PART 2

Planning a JNU Implementation

- Planning Overview on page 9

CHAPTER 2

Planning Overview

- Platform Considerations for the JNU Controller on page 9
- Platform Considerations for JNU Satellite Devices on page 9

Platform Considerations for the JNU Controller

You can use any MX Series 3D Universal Edge Router as the JNU controller. The MX Series router uses Modular Port Concentrators (MPCs) to connect to the satellites.

You must use an MX Series router as the controller, and you must use MPCs (not DPCs). We recommend that you allocate more than one interface for interconnect between MX Series routers and satellites. These interfaces will be placed into LAG configuration for fast recovery, with traffic spreading across member links.

An MX Series router can manage one satellite on each of its Ethernet ports. For example, the MX 960 router supports up to 176 10-Gigabit Ethernet interfaces. It can therefore manage up to 176 satellite devices on the 10-Gigabit Ethernet interfaces.

Platform Considerations for JNU Satellite Devices

JNU 1.0 supports the following satellite devices:

- EX3300 Ethernet Switches
- EX4200 Ethernet Switches
- QFX 3500 devices

Table 2 on page 10 gives an overview of these devices.

Table 2: Comparison of Satellite Devices That Support JNU

Model	Typical Deployment Needs	Ports	Power Supply/Fans	Uplink Ports	Power over Ethernet (PoE) Ports	Airflow Direction	Form Factor	Virtual Chassis Support
EX3300 switch	Scalable connectivity for the enterprise market, including branch offices, campus locations, and data centers	24 or 48 ports with 10/100/1000BASE-T Gigabit Ethernet connectors	Fixed AC or DC power supplies	4 autosensing uplink ports that support small form-factor pluggable (SFP) and SFP+ transceivers	PoE+ on some models	Front-to-back Back-to-front	12 in deep (about 1 U)	6-member virtual chassis Virtual Chassis over 10-Gigabit Ethernet uplinks
EX4200 switch	High density of Gigabit Ethernet ports or redundancy. Used in large branch offices, campus wiring closets, and data centers	24 or 48 ports with 10/100/1000BASE-T Gigabit Ethernet connectors	Dual redundant AC/DC power supplies that are field-replaceable and hot-swappable. Optional connection to external power source. Field-replaceable fan tray with three fans.	• 1-gigabit SFP transceivers • 10-gigabit SFP+ transceivers • 10-gigabit XFP transceivers	PoE and PoE+ on 8, 24, or 48 ports	Side-to-back	?	10-member virtual chassis Virtual Chassis over 10-Gigabit Ethernet uplinks
QFX3500 device	10GE scale or mixed GE/10GE scale	4x10GE or 4 * 40G future uplink ports (can also operate in 16x10G mode)		48 10-Gbps access ports use SFP+ transceivers and operate by default as 10-Gigabit Ethernet interfaces		Front-to-back	1 rack unit (1 U)	

PART 3

Implementing JNU

- Getting Started with the JNU Software on page 13
- Using SNMP and System Logging with JNU on page 27
- Configuring Junos OS Features with JNU on page 31
- Committing Configurations on page 35
- JNU Operational Mode Commands on page 37
- Setting Up a Basic JNU Implementation on page 55

CHAPTER 3

Getting Started with the JNU Software

- Installing the JNU Software on the Controller on page 13
- Installing the JNU Software on the Satellite Devices on page 13
- Initializing JNU Mode on the Controller on page 14
- Initializing JNU Mode on the Satellite Devices on page 21

Installing the JNU Software on the Controller

To load the JNU package onto the controller:

- Enter the following command on the MX Series controller:

```
user@jnu1-ctrlr> request system software add jnu-1.0R1.0-signed.tgz
Installing package '/var/tmp/jnu-1.0R1-signed.tgz' ...
Verified jnu-1.0R1.tgz signed by PackageProduction_11_4_0 Adding jnu...
Available space: 556676 require: 3220
NOTICE: uncommitted changes have been saved in
/var/db/config/juniper.conf.pre-install
Mounted jnu package on /dev/md10...
Restarting bslockd ...
mgd: commit complete
Saving package file in /var/sw/pkg/jnu-1.0R1-signed.tgz ...
Saving state for rollback ...
```

Installing the JNU Software on the Satellite Devices

To load the JNU package onto the satellite device:

- Enter the following command on the satellite device.

```
user@jnu-satellite1> request system software add jnu-1.0R1.0-signed.tgz
Installing package '/var/tmp/jnu-1.0R1-signed.tgz' ...
Verified jnu-1.0R1.tgz signed by PackageProduction_11_4_0 Adding jnu...
Available space: 556676 require: 3220
NOTICE: uncommitted changes have been saved in
/var/db/config/juniper.conf.pre-install
Mounted jnu package on /dev/md10...
Restarting bslockd ...
mgd: commit complete
Saving package file in /var/sw/pkg/jnu-1.0R1-signed.tgz ...
Saving state for rollback ...
```

Initializing JNU Mode on the Controller

After you install the JNU software, you need to initially configure and initialize the MX Series controller. This example configures the controller, adds two satellite devices to the controller configuration, and configures SNMP, system logging, and NTP on the controller.

The first time you initialize the controller, you must enter the full command **op url /var/db/scripts/op/initialize-controller.slax**. Thereafter, you can re-initialize the controller using the **op jnu-initialize-controller** command.

For a description of the fields used to initialize the controller, see jnu-initialize-controller.

To initially configure the controller:

1. Enter the **initialize-controller** command and follow the prompts.

```
user@jnu1-ctrlr> op url /var/db/scripts/op/initialize-controller.slax
Controller initializations:
  Please enter hostname [jnu-controller]:
  Please enter management IP prefix [192.168.0.1/24]:
  Please enter management VLAN id [4094]:
  Do you want to configure any satellites now [n]: y

    Please enter the number of satellites [1]: 2
    Satellite 1
      Please enter the hostname of the satellite: jnu-sat1
      Please enter the IP address of the satelltie: 192.168.0.2
      Please enter downlink interfaces to satellite: ge-0/0/0
    Satellite 2
      Please enter the hostname of the satellite: jnu-sat2
      Please enter the IP address of the satelltie: 192.168.0.3
      Please enter downlink interfaces for satellite: ge-0/0/1

  Do you want to configure SNMP [n]: y
   Do you want to enter a read-only community string (y|n)? y
      SNMP read-only community string: public
    Do you want to enter a read-only community string (y|n)? y
      SNMP read-only community string: private
    Do you want to enter SNMP trap parameters (y|n)? y
      SNMP trap target address: 169.37.0.1
    Do you want to enter SNMP trap categories (y|n)? y
    Do you want to enable SNMP trap for 'otn-alarms' (y|n)? y
      Available alarms:
        'oc-lof', 'oc-lom', 'oc-los', 'odu-ais', 'odu-bbe-threshold',
        'odu-bdi', 'odu-es-threshold', 'odu-lck', 'odu-oci',
        'odu-rx-aps-change', 'odu-sd', 'odu-ses-threshold', 'odu-sf',
        'odu-ttim', 'odu-uas-threshold', 'opu-ptm', 'otu-ais',
        'otu-bbe-threshold', 'otu-bdi', 'otu-es-threshold',
        'otu-fec-deg', 'otu-fec-exe', 'otu-iae', 'otu-sd',
        'otu-ses-threshold', 'otu-sf', 'otu-ttim', 'otu-usa-threshold',
        'wavelength-lock'
      Please enter otn-alarms: oc-lof,oc-lom
    Do you want to enable SNMP trap for 'sonet-alarms' (y|n)? y
      Available alarms:
       'ber-defect', 'ber-fault', 'line-ais', 'line-remote-defect-indication',

        'loss-of-cell', 'loss-of-frame', 'loss-of-light', 'loss-of-pointer',
        'loss-of-signal', 'path-ais', 'path-mismatch',
       'path-remote-defect-indication', 'pll-lock', 'remote-error-indication',

        'severely-errored-frame', 'unequipped', 'vt-ais', vt-label-mismatch',

       'vt-loss-of-cell', 'vt-loss-of-pointer', 'vt-remote-defect-indication',

        'vt-unequipped'
      Please enter sonet-alarms: path-ais
    Other categories:
      'authentication', 'chassis', 'configuration', 'link',
      'remote-operations', 'rmon-alarm', 'routing', 'services',
      'startup', 'vrrp-events'
    Do you want to enter other SNMP trap categories (y|n)? y
      Please enter SNMP trap categories: vrrp-events
  Do you want to configure Syslog server [n]: y
Syslog host address? 167.37.0.1
```

```
        port number [123]:
    Sylog facility 'all' [n]:
    Syslog facilities:
      'authorization', 'change-log',        'conflict-log', 'daemon',
      'dfc',           'explicit-priority', 'external',     'firewall',
      'ftp',           'interactive-commands', 'kernel',    'log-prefix',
      'ntp',           'pfe',               'security',     'user'
    Syslog severities:
      'alert', 'any',  'criticial', 'emergency', 'error',
      'info',  'none', 'notice',    'warning'
    Please enter syslog facility name: change-log
    Please enter severity: warning
    Do you want to enter more syslog facilities [n]?
  Do you want to configure NTP [n]: y
    NTP server address: 168.37.0.1


JNU controller configuration completed
```

As part of the initialization process, the JNU configuration is committed on the controller.

The following is an example of the configuration loaded onto the controller as a result of running the controller initialization process.

```
chassis {
  aggregated-devices {
    ethernet {
      device-count 480;
    }
  }
  /* Slot of the Trio FPC */
  fpc 5 {
    pic 0 {
      inline-services {
        bandwidth 1g;
      }
    }
  }
}
  system {
    ntp {
      /* The server is in the main routing-instance, */
      /* the server parameters will not be propagated to the satellites */
      server 188.88.0.1; /* external server */
    }
    syslog {
      host 169.37.0.2 {
        security info;
        change-log info;
        /* All the syslog parameters are propagated to satellite */
        /* except source-address. The source address used by */
        /* the satellites are the mgmt address on satellite */
        source-address 137.34.1.1;
      }
      file messages {
        any any;
      }
```

```
    }
    services {
        ssh;
    }
}
snmp {
    /* community is configured by user, satellite community string is */
    /* <controller_community_string>:<satellite-name> */
    community public {
        authorization read-only;
    }
    community private {
        authorization read-write;
    }
    trap-options {
        source-address lo0;
    }
    trap-group public {
        /* categories are configured by user, propagate to satellite */
        categories {
            authentication;
            routing;
        }
        /* targets are configured by user, propagate to satellite */
        targets {
            169.37.0.1;
        }
    }
    /* Need SNMP proxy configuration */
    proxy jnu-sat1 {
        device-name 192.168.0.2;
        version-v2c {
            snmp-community public:jnu-sat1;
        }
        routing-instance jnu-vrf;
    }
    proxy jnu-sat2 {
        device-name 192.168.0.3;
        version-v2c {
            snmp-community public:jnu-sat2;
        }
        routing-instance jnu-vrf;
    }
}
security {
    ssh-known-hosts {
        host 192.168.0.2 {
            ecdsa-sha2-nistp256-key
                [illegible]
        }
        host 192.168.0.3 {
            ecdsa-sha2-nistp256-key
                [illegible]
        }
    }
}
```

```
interfaces {
  lo0 {
    /* The loopback interface is not part of JNU configuration */
    /* Just an illustration of where the management IP address comes from */
    unit 0 {
      family inet {
        address 137.34.1.1/32;
      }
    }
  }
  /* All the interfaces connecting to satellites */
  ge-0/0/0 {
    gigether-options {
      802.3ad ae479;
    }
  }
  si-5/0/0 {
    unit 0 {
      family inet;
      family inet6;
    }
    unit 1 {
      family inet;
      service-domain inside;
    }
    unit 2 {
      family inet;
      service-domain outside;
    }
  }
  ae479 {
    /* Using aggregate-ethernet interface because there can be */
    /* multiple physical downlinks */
    aggregated-ethernet-options {
      lacp {
        active;
      }
    }
    vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 16385 {
      encapsulation vlan-bridge;
      vlan-id 4094;
    }
  }
  irb {
    unit 16385 {
      family inet {
        address 192.168.0.1/24;
      }
    }
  }
}
policy-options {
  policy-statement reject-all {
    then reject;
```

```
    }
  }
  routing-instances {
    jnu-vrf {
      instance-type vrf;
      interface irb.16385;
      interface si-5/0/0.1;
      route-distinguisher 192.168.0.1:0;
      vrf-import reject-all;
      vrf-export reject-all;
      routing-options {
        static {
          /* Static route to SNMP trap server via si- interface */
          route 169.37.0.1/32 next-hop si-5/0/0.1;
          /* Static route to syslog server via si- interface */
          route 169.37.0.2/32 next-hop si-5/0/0.1;
        }
      }
    }
    jnu-vs {
      instance-type virtual-switch;
      bridge-domains {
        jnu {
          vlan-id 4094;
          interface ae479.16385;
          routing-interface irb.16385;
        }
      }
    }
  }
  services {
    service-set ss-nat {
      nat-rules jnu-use-controller;
      next-hop-service {
        inside-service-interface si-5/0/0.1;
        outside-service-interface si-5/0/0.2;
      }
    }
    nat {
      /* There needs to be 1 NAT pool (with the same address) per satellite */
      pool jnu-sat1 {
        /* Use Management IP address */
        address 137.34.0.1/32;
      }
      pool jnu-sat2 {
        /* Use Management IP address */
        address 137.34.0.1/32;
      }
      allow-overlapping-nat-pools;
      rule jnu-use-controller {
        match-direction input;
        term jnu-sat1 {
          /* Each satellite connection will use 1 term on its own */
          from {
            source-address {
              /* 1st satellite */
```

```
                10.0.0.1/32;
              }
            }
            then {
              translated {
                source-pool jnu-sat1;
                translation-type {
                  basic-nat44;
                }
              }
            }
          }
          term jnu-sat2 {
            /* Each satellite connection will use 1 term on its own */
            from {
              source-address {
                /* 2nd satellite */
                10.0.0.2/32;
              }
            }
            then {
              translated {
                source-pool jnu-sat2;
                translation-type {
                  basic-nat44;
                }
              }
            }
          }
        }
      }
    }
```

Initializing JNU Mode on the Satellite Devices

When you initialize the satellite device, the software creates a management configuration on the satellite device that allows the controller to configure and manage the satellite.

When you run the satellite initialization process, the controller connects to the satellite and copies JNU code elements that are based on scripting technology to the satellite.

The initialization process loads an SSH public key onto the satellite devices and sets up a NETCONF connection over SSH for use between the controller and the satellite devices. It creates a user account called JNU-admin, which the controller uses to log in to the satellites.

Before you initialize the satellite device, you must configure a root (superuser) password by including the **root-authentication** statement at the **[edit system]** hierarchy level

The first time you initialize the controller, you must enter the full command **op url /var/db/scripts/op/initialize-satellite.slax**. Thereafter, you can re-initialize the controller using the **op jnu-initialize-controller** command.

To initialize a satellite device:

1. Enter the following command on the satellite device, and follow the prompts.

```
user@jnu-satellite1> op url /var/db/scripts/op/initialize-satellite.slax
Satellite initializations:
  Please enter hostname [jnu-satellite1]: jnu1-sat-ex1

  Please enter management IP address: 10.0.0.1

  Please enter uplink IP prefix (192.168.0.2-254/24): 192.168.0.2/24

  Please enter uplink interface name: ge-0/0/0
  Please enter management VLAN id [4094]:
  Please enter controller IP address [192.168.0.1]:
```

As part of the initialization process, the JNU configuration is committed on the satellite device.

The following is an example of the configuration that is initialized on the satellite device. To display this configuration from the MX Series controller, enter the following command:

```
user@jnu1-ctrlr> op jnu-show-configuration device jnu1-sat-ex1 source committed
## Last commit: 2012-07-23 08:59:09 UTC by jnuadmin
version 11.4R4.2;
/*
 * $Id$
 *
 * ex4200-defaults.conf  - Default configurations for EX4200
 *
 * Copyright (c) 2010, Juniper Networks, Inc.
 * All rights reserved.
 */
```

The following output is from an EX4200 Ethernet Switch.

```
groups {
    global {
        system {
            root-authentication {
                encrypted-password LsaElrdR.Y7qs;
            }
        }
    }
    member0 {
        interfaces {
            me0 {
                unit 0 {
                    family inet {
                        address 192.168.164.83/22;
                    }
                }
            }
        }
    }
    jnu1-sat-ex1-freeform;
    jnu-module {
        system {
            scripts {
                commit {
                    allow-transients;
                    /* JNU configuration templates */
                    file config-analyzer-commit-ex4200.slax;
                    file config-vstp-commit-ex4200.slax;
                    file config-vrrp-commit-ex4200.slax;
                    file config-system-internet-options-commit-ex4200.slax;
                    file config-system-accounting-commit-ex4200.slax;
                    file config-stp-commit-ex4200.slax;
                    file config-sflow-commit-ex4200.slax;
                    file config-rstp-commit-ex4200.slax;
                    file config-routing-options-static-commit-ex4200.slax;
                    file config-routing-options-rib-groups-commit-ex4200.slax;
                    file config-routing-options-martians-commit-ex4200.slax;
                    file config-routing-options-generate-commit-ex4200.slax;
                    file config-routing-options-flow-commit-ex4200.slax;
                    file config-routing-options-commit-ex4200.slax;
                    file config-routing-options-aggregate-commit-ex4200.slax;
                    file config-routing-options-access-internal-commit-ex4200.slax;
                    file config-routing-options-access-commit-ex4200.slax;
                    file config-policy-options-prefix-list-commit-ex4200.slax;
                    file config-policy-options-damping-commit-ex4200.slax;
                    file config-policy-options-condition-commit-ex4200.slax;
                    file config-policy-options-community-commit-ex4200.slax;
                    file config-policy-options-as-path-commit-ex4200.slax;
                    file config-poe-commit-ex4200.slax;
                    file config-ospf-commit-ex4200.slax;
                    file config-mstp-commit-ex4200.slax;
                    file config-lldp-med-commit-ex4200.slax;
                    file config-lldp-commit-ex4200.slax;
                    file config-link-management-commit-ex4200.slax;
                    file config-lacp-commit-ex4200.slax;
                    file config-l2vpn-commit-ex4200.slax;
```

```
                file config-interface-commit-ex4200.slax;
                file config-free-form-commit-ex4200.slax;
                file config-firewall-three-color-policer-commit-ex4200.slax;
                file config-firewall-policer-commit-ex4200.slax;
                file config-ethsw-storm-control-commit-ex4200.slax;
                file config-ethsw-secure-access-port-commit-ex4200.slax;
                file config-ethsw-redundant-trunk-group-commit-ex4200.slax;
                file config-ethsw-options-commit-ex4200.slax;
                file config-ethsw-bpdu-commit-ex4200.slax;
                file config-dot1x-commit-ex4200.slax;
                file config-cos-schedulers-commit-ex4200.slax;
                file config-cos-rewrite-rules-commit-ex4200.slax;
                file config-cos-host-outbound-traffic-commit-ex4200.slax;
                file config-cos-forwarding-class-commit-ex4200.slax;
                file config-cos-drop-profiles-commit-ex4200.slax;
                file config-cos-classifiers-commit-ex4200.slax;
                file config-cos-code-point-alias-commit-ex4200.slax;
            }
            op {
                file jnu-initialize-satellite.slax;
                file load-base-conf-ex4200.slax;
                file get-and-load-conf-ex4200.slax;
                file config-free-form-precommit.slax;
            }
        }
    }
}
}
apply-groups [ member0 global jnu1-sat-ex1-freeform jnu-module ];
    system {
        host-name jnu1-sat-ex1;
        login {
            user jnuadmin {
                uid 928;
                class super-user;
                shell csh;
                authentication {
                    encrypted-password "$1$a7DYfEN0$huf8FDLIshFB5v6zb2WMz0";
                }
            }
        }
        syslog {
            file messages {
                any any;
            }
        }
    }
    chassis {
        aggregated-devices {
            ethernet {
                device-count 63;
            }
        }
    }
    interfaces {
        /* Uplink interface to controller; Using aggregated Ethernet because there can be more
```

```
    than 1 uplink */
    xe-0/1/0 {
      ether-options {
        802.3ad ae62;
      }
    }
    ae62 {
      /* Aggregated Ethernet interface uplink connection to controller */
      vlan-tagging;
      aggregated-ether-options {
        lacp {
          active;
        }
      }
      unit 16385 {
        vlan-id 4094;
        family inet {
          address 192.168.0.2/24;
        }
      }
    }
}
event-options {
    generate-event {
      event-script-timer time-interval 300;
    }
    policy jnu-controller-connectivity {
      events event-script-timer;
      then {
        event-script monitor-controller-ex4200.slax {
          arguments {
            cntrlr-ip 192.168.0.1;
          }
        }
      }
    }
    event-script {
      file monitor-controller-ex4200.slax;
    }
}
routing-options {
    static {
      rib-group ntp;
    }
    rib-groups {
      ntp {
        import-rib [ inet.0 jnu.inet.0 ];
        import-policy jnu-mgmt;
      }
    }
}
policy-options {
    policy-statement jnu-mgmt {
      from protocol static;
      then accept;
    }
```

```
    policy-statement reject-all {
        then reject;
    }
}
routing-instances {
    jnu {
        instance-type vrf;
        interface ae62.16385;
        route-distinguisher 192.168.0.2:0;
        vrf-import reject-all;
        vrf-export reject-all;
        routing-options {
            static {
                route 0.0.0.0/0 next-hop 192.168.0.1;
            }
        }
    }
}
```


CHAPTER 4

Using SNMP and System Logging with JNU

- Centralized Collection of SNMP Statistics and Log Messages on page 27
- Overview of Using the JNU Controller as an SNMP Proxy Agent on page 27
- Configuring the JNU Controller as an SNMP Proxy Agent on page 28

Centralized Collection of SNMP Statistics and Log Messages

The JNU software provides a single point of collecting SNMP statistics and sending SNMP traps to the SNMP server. Network Management System (NMS) polls the controller for SNMP statistics on the controller and the satellite devices. SNMP statistics from the satellite devices are routed through the controller. A Network Address Translation (NAT) configuration set up by the controller initialization script is used to translate the source address of the satellite devices to the source address of the controller for traffic sent to the SNMP server. This process means that all SNMP traffic originates from one source address on the controller.

The same community strings (SNMPv2) and context engine ID (SNMPv3) assigned to the controller and each of the satellite devices are used by these systems to send the SNMP trap messages, so that the NMS can differentiate them.

Collecting Log Messages

The JNU software provides a single point of collecting logging messages and sending them to the syslog server. The controller sends all syslog messages for the JNU group of satellites and controller to the server. The NAT configuration set up by the controller initialization script is used to translate the source address of the satellite devices to the source address of the controller for traffic sent to the syslog server. This process means that all logging traffic originates from one source address on the controller.

Overview of Using the JNU Controller as an SNMP Proxy Agent

The Junos OS SNMP proxy feature allows you to set up the controller as a proxy SNMP agent through which the network management system (NMS) can query satellite devices.

When the JNU controller acts as the proxy SNMP agent for the satellite devices, the NMS specifies the community name (for SNMPv1 and SNMPv2) or the context and security

name (for SNMPv3) of the satellite device from which it requires the information. If you have configured authentication and privacy methods and passwords for SNMPv3, those parameters are also specified in the query for SNMPv3 information.

Configuring the JNU Controller as an SNMP Proxy Agent

The community and security configuration for the proxy should match the corresponding configuration on the device that is to be managed.

Because the proxy SNMP agent does not have trap forwarding capabilities, the devices that are managed by the proxy SNMP agent send the traps directly to the network management system.

To configure the satellite as an SNMP proxy agent:

1. Create a proxy configuration, and assign a name to the configuration.

   ```
   user@jnu1-ctrlr# edit snmp proxy proxy-ctrlr
   ```

2. Assign the proxy configuration to a satellite device.

   ```
   [edit snmp proxy proxy-ctrlr]
   user@jnu1-ctrlr# set device-name jnu-sat-1
   ```

3. (Optional) Configure SNMP version 1. Configure a community string in the format *controller-community-string*:*satellite-hostname*.

   ```
   [edit snmp proxy proxy-ctrlr]
   user@jnu1-ctrlr# edit version-v1
   [edit snmp proxy proxy-ctrlr version-v1]
   user@jnu1-ctrlr# set snmp-community public:jnu-sat-1
   ```

   You can optionally enter the **no-default-comm-to-v3-config** statement. If you include it in the configuration, you must manually configure the statements at the **[edit snmp v3 snmp-community community-name]** and **[edit snmp v3 vacm]** hierarchy levels. If you do not include the statement, the **[edit snmp v3 snmp-community community-name]** and **[edit snmp v3 vacm]** hierarchy level configurations are automatically initialized.

4. (Optional) Configure SNMP version 2c. Configure a community string in the format *controller-community-string*:*satellite-hostname*.

   ```
   [edit snmp proxy proxy-ctrlr]
   user@jnu1-ctrlr# edit version-v2c
   [edit snmp proxy proxy-ctrlr version-v2c]
   user@jnu1-ctrlr# set snmp-community public:jnu-sat-1
   ```

   You can optionally enter the **no-default-comm-to-v3-config** statement. If you include it in the configuration, you must manually configure the statements at the **[edit snmp v3 snmp-community community-name]** and **[edit snmp v3 vacm]** hierarchy levels. If you do not include the statement, the **[edit snmp v3 snmp-community community-name]** and **[edit snmp v3 vacm]** hierarchy level configurations are automatically initialized.

5. (Optional) Configure SNMP version 3. Specify a security name to be used for messaging security and user access control. Specify the ID of the SNMP context that is accessible to the SNMP proxy.

   ```
   [edit snmp proxy proxy-ctrlr]
   user@jnu1-ctrlr# edit version-v3
   [edit snmp proxy proxy-ctrlr version-v3]
   user@jnu1-ctrlr# set security-name jnu-user
   user@jnu1-ctrlr#  set context jnu
   ```

   NOTE: This security name must match the security name configured at the **[edit snmp v3 target-parameters *target-parameters-name* parameters]** hierarchy level when you configure traps.

6. (Optional) Create this SNMP proxy for a logical system on the satellite device.

   ```
   [edit snmp proxy proxy-ctrlr]
   user@jnu1-ctrlr# set logical-system ls1
   ```

7. (Optional) Create this SNMP proxy for a routing instance on the satellite device.

   ```
   [edit snmp proxy proxy-ctrlr]
   user@jnu1-ctrlr# set routing-instance rs1
   ```

You can use the **show snmp proxy** operational mode command to view proxy details on a device. The **show snmp proxy** command returns the proxy names, device names, SNMP version, community/security, and context information

CHAPTER 5

Configuring Junos OS Features with JNU

- Configuring Junos Features with JNU Configuration Templates on page 31
- Configuring Junos Features with JNU Free Form on page 32

Configuring Junos Features with JNU Configuration Templates

The JNU software comes with configuration templates that you can use to configure Junos OS features. Each template contains parameters that correspond to a set of Junos OS configuration statements. You configure these parameters with the same values that you would use for the corresponding **set** statement in configuration mode of the Junos OS CLI. After you have finished configuring the templates, run the **op jnu-commit** command to commit the new configuration on the specified satellite devices.

Displaying a List of Configuration Templates

To see a list of templates, enter **op config-?** In operational mode. For example:

```
user@jnu1-ctrlr> op config-?
Possible completions:

<script>
 config-analyzer
  config-cos-classifiers
  config-cos-code-point-alias
  config-cos-congestion-notification-profile
  config-cos-drop-profiles

. . .

  config-system-internet-options
  config-system-login
  config-system-syslog
  config-vlan
  config-vrrp
  config-vstp
```

Displaying the Configuration Parameters in a Template

You can display a list of parameters in a template. If there is a range of accepted values or a particular value accepted for a parameter, these are included in parenthesis. To display a list of parameters in a template, enter the name of the template with a **?** . For example:

```
user@jnu1-ctrlr> op config-cos-drop-profiles ?
Possible completions:

  action               Action to be performed ('create', 'delete')
  apply-groups         Groups from which to inherit configuration data
  apply-groups-except  Don't inherit configuration data from these groups
  detail               Display detailed output
  device               Controller/Satellite Name
  drop-profile.name    Random Early Drop (RED) data point map
  fill-level           Fill-level value of data point (0 .. 100 percent)
  fill-level.drop-probability  Probability packet will be dropped
  group                Configuration group name
  interpolate.drop-probability  Data points for packet drop probability (0 .. 100
 percent)
  interpolate.fill-level  Data points for queue full percentage (0 .. 100 percent)

  |                    Pipe through a command
```

Configuring the Template

To configure a template:

- Include the **device** command, which specifies the satellite device on which you want to commit the configuration. You can configure only one satellite device at a time using the configuration templates.
- Include the **action** command, which specifies whether you are creating a configuration or deleting a configuration.
- Add parameters and values on one line in any order. The software does not validate values, but it notifies you if you miss a required parameter.

For example, to create a drop policy called best-effort on the **jnu1-sat-ex1** satellite device:

```
user@jnu1-ctrlr> op config-cos-drop-profiles drop-profile.name best-effort
  interpolate.fill-level 30 fill-level 50 fill-level.drop-probability 0
  interpolate.drop-probability 80 device jnu1-sat-ex1 action create
```

Committing the Configuration

After you have finished configuring the templates, run the **jnu-commit** command to commit the new configuration on the specified satellite devices.

Configuring Junos Features with JNU Free Form

You can use the **config-free-form** command to configure Junos OS **set** statements on satellite devices. You can configure any **set** statement that is supported on the satellite device.

To use the **config-free-form** command to configure Junos OS **set** statements:

- Include the **action** command, which specifies whether you are adding a statement or deleting a statement.

- Include the **device** command, which specifies one or more satellite devices on which you want to commit the configuration. Enter multiple satellite devices in a comma-separated list.
- Add statements and values on one line in any order. The software does not validate values, but it notifies you if you miss a required parameter.

```
op config-free-form action add device jnu1-sat-ex1 command "set interfaces address
   10.10.1.1 family inet unit 0 vlan-id 1044"
set interfaces ge-1/0/0 unit 0 vlan-id 1044 family inet address 10.10.1.1
set routing-options static route 172.16.0.0 next-hop 192.168.167.254 retain no-readvertise
```

CHAPTER 6

Committing Configurations

- Commit Process for Satellites Already Connected to the Controller on page 35
- Commit Process for Satellite Devices That Come Online After the Commit Process on the Controller on page 36

Commit Process for Satellites Already Connected to the Controller

JNU uses commit scripts to automate the commit process on the controller and satellite devices. You commit configurations for the satellite devices from the controller. You should modify and commit configuration changes for satellite devices only on the controller.

When you commit a configuration on the controller, the flow of the commit process on the controller is as follows:

1. Enter the following command on the controller:

   user@jnu-ctrlr> **op jnu-commit**

2. The controller polls each satellite device to verify that the device is reachable.
3. The controller sends the new satellite configuration to each satellite device by using the NETCONF XML management protocol.
4. The controller runs the remote procedure call (RPC) validate process on each satellite device to validate the new configuration.
5. If all the satellite devices successfully validate the new configuration, the controller runs the commit script, which runs the RPC commit process on all satellite devices.

   When the process is complete, the controller displays the following message:

   ```
   jnu1-sat-ex1:
   Configuration check succeeds
   jnu1-sat-qfx2:
   Configuration check succeeds
   ```

   If the new configuration is not successfully validated on all satellite devices in the JNU network, the commit process stops and the controller displays an error message.

Commit Process for Satellite Devices That Come Online After the Commit Process on the Controller

If a satellite device is not connected to the controller when you perform the commit process, it receives its configuration from the controller when it comes online.

The commit process for satellite devices that come online after the commit process is complete:

1. The satellite device comes online, and the satellite and the controller discover the other.
2. A management connection is made between the controller and the satellite device by means of Junos OS automation features.
3. If the controller has a new configuration for the satellite device, the controller sends the new configuration to the satellite device.
4. The satellite device validates the configuration, and if the validation succeeds, the configuration is committed on the satellite.

   The configuration on the satellite is now synchronized with the rest of the JNU group.

If the commit process fails or if the controller does not have a new configuration for the satellite device, the controller removes any services previously committed on the satellite device because the configuration will not be synchronized the rest of the JNU group. The controller restores the configuration that the satellite device had in discovery mode. Because the satellite device has an open management channel to the controller, it will participate in subsequent configuration commits that the controller sends.

CHAPTER 7

JNU Operational Mode Commands

jnu-add-delete-satellites

Syntax

```
op
  jnu-add-delete-satellites
  action add | delete
    downlink interface-id
    inet ip-address
    satellite satellite-name
```

Release Information Command introduced in JNU 1.0.

Description Add or delete satellite devices in the controller configuration.

Options

action add | delete—Adds or deletes the specified satellite device.

downlink *interface-id*—ID of the interface used for the downlink connection to the satellite device.

inet *ip-address*—IP address used to communicate with the satellite device.

satellite *satellite-name*—Name of the satellite device to be added or deleted.

config-free-form

Syntax

```
op
  config-free-form
    "set statement-name value"
    device device-name
    action (add | delete)
```

Release Information Command introduced in JNU 1.0.

Description Configure Junos OS set statements on the satellite device. You can use **config-free-form** to configure any set statement that is supported on the satellite device. You can enter the statements in any order on the same line.

Options

"set *statement-name value*"—A **set** statement and value that is supported on the satellite device. You can enter multiple **set** statement on the same line. The JNU software does not validate the value that you enter for the statement. The value is validated during the commit process.

device *device-name*—Device to which you want to add or remote the statement

action (add | delete)—Action to be taken on the **set** statements. You can add the statement to the configuration of the satellite device or remove the set statement from the configuration of the satellite device.

jnu-initialize-controller

Syntax
op
jnu-initialize-controller

Release Information Command introduced in JNU 1.0.

Description The first time you initialize JNU on the controller, you must run the **op url /var/db/scripts/op/initialize-controller** operational mode command. Thereafter, to change the JNU configuration on the controller, you can run the **op jnu-initialize-controller** command.

Output Fields Table 3 on page 40 describes the fields that you fill in when you run the **op jnu-initialize-controller** command.

Table 3: jnu-initialize-controller Output Fields

Field	Description
Please enter hostname	Hostname for the controller.
Please enter management IP prefix [192.168.0.1/24]	IP address of the controller used to manage the satellite devices.
Please enter management VLAN id [4094]	VLAN ID used on the management network.
Do you want to configure any satellites now	Enter **y** to add satellite devices to the configuration.
Please enter the number of satellites	Number of satellite devices that you want to add to the configuration.
Please enter the hostname of the satellite	Name for the satellite.
Please enter the IP address of the satellite	IP address for the satellite. This is the address that the controller uses to manage the satellite.
Please enter downlink interfaces to satellite	Interface ID of the satellite. This is the interface that the controller uses to manage the satellite.
Do you want to configure SNMP [n]	Enter **y** to configure SNMP on the controller.
Do you want to enter a read-only community string (y | n)	Enter **y** to configure a community string for the controller, and then enter the community string. Repeat this process to enter a second community string.
Do you want to enter SNMP trap parameters (y | n)?	Enter **y** to configure SNMP traps.
SNMP trap target address:	If you entered **y** to configure SNMP traps, specify the address to which traps are sent.
Do you want to enter SNMP trap categories (y | n)?	Enter **y** to specify trap categories.

Table 3: jnu-initialize-controller Output Fields *(continued)*

Field	Description
Do you want to enable SNMP trap for 'otn-alarms' (y | n)?	Enter **y** to specify Optical Transport Network (OTN) alarm categories. A list of available alarms displays. Enter alarms in a comma-separated list.
Do you want to enable SNMP trap for 'sonet-alarms' (y | n)?	Enter **y** to specify SONET/SDH alarm categories. A list of available alarms displays. Enter alarms in a comma-separated list.
Other categories:	A list of additional trap categories that you can add to the trap configuration displays.
Do you want to enter other SNMP trap categories (y | n)?	Enter **y** to add other trap categories to your configuration.
Please enter SNMP trap categories:	Comma-separated list of additional trap categories that you want to add to your configuration.
Do you want to configure Syslog server [n]:	Enter **y** to configure a server to which system log messages are sent.
Syslog host address?	IP address of the system logging server.
port number [123]:	Port number of the system logging server.
Sylog facility 'all' [n]:	Class of messages to log. Enter **y** to enable all facilities; enter **n** to display a list of facilities that you can add to the configuration. .
Syslog facilities:	List of facilities that you can add to the configuration.
Syslog severities:	List of message severities.
Please enter syslog facility name:	Name of the facilitiy that you want to add to the configuration.
Please enter severity:	Specify the severity of the messages that belong to the facility. Messages with severities of the specified level and higher are logged.
Do you want to enter more syslog facilities [n]?	Enter **y** to configure additional facilities.
Do you want to configure NTP [n]:	Enter y to configure an NTP server.
NTP server address:	IP address of the NTP server.

Sample Output

```
user@jnu1-ctrlr> op initialize-controller
Controller initializations:
```

```
  Please enter hostname [jnu-controller]:
  Please enter management IP prefix [192.168.0.1/24]:
  Please enter management VLAN id [4094]:
  Do you want to configure any satellites now [n]: y

    Please enter the number of satellites [1]: 2
    Satellite 1
      Please enter the hostname of the satellite: jnu-sat1
      Please enter the IP address of the satellite: 192.168.0.2
      Please enter downlink interfaces to satellite: ge-0/0/0
    Satellite 2
      Please enter the hostname of the satellite: jnu-sat2
      Please enter the IP address of the satelltie: 192.168.0.3
      Please enter downlink interfaces for satellite: ge-0/0/1

  Do you want to configure SNMP [n]: y
   Do you want to enter a read-only community string (y|n)? y
      SNMP read-only community string: public
    Do you want to enter a read-only community string (y|n)? y
      SNMP read-only community string: private
    Do you want to enter SNMP trap parameters (y|n)? y
      SNMP trap target address: 169.37.0.1
    Do you want to enter SNMP trap categories (y|n)? y
    Do you want to enable SNMP trap for 'otn-alarms' (y|n)? y
      Available alarms:
        'oc-lof', 'oc-lom', 'oc-los', 'odu-ais', 'odu-bbe-threshold',
        'odu-bdi', 'odu-es-threshold', 'odu-lck', 'odu-oci',
        'odu-rx-aps-change', 'odu-sd', 'odu-ses-threshold', 'odu-sf',
        'odu-ttim', 'odu-uas-threshold', 'opu-ptm', 'otu-ais',
        'otu-bbe-threshold', 'otu-bdi', 'otu-es-threshold',
        'otu-fec-deg', 'otu-fec-exe', 'otu-iae', 'otu-sd',
        'otu-ses-threshold', 'otu-sf', 'otu-ttim', 'otu-usa-threshold',
        'wavelength-lock'
      Please enter otn-alarms: oc-lof,oc-lom
    Do you want to enable SNMP trap for 'sonet-alarms' (y|n)? y
      Available alarms:
        'ber-defect', 'ber-fault', 'line-ais', 'line-remote-defect-indication',
        'loss-of-cell', 'loss-of-frame', 'loss-of-light', 'loss-of-pointer',
        'loss-of-signal', 'path-ais', 'path-mismatch',
        'path-remote-defect-indication', 'pll-lock', 'remote-error-indication',
        'severely-errored-frame', 'unequipped', 'vt-ais', vt-label-mismatch',
        'vt-loss-of-cell', 'vt-loss-of-pointer', 'vt-remote-defect-indication',
        'vt-unequipped'
      Please enter sonet-alarms: path-ais
    Other categories:
      'authentication', 'chassis', 'configuration', 'link',
      'remote-operations', 'rmon-alarm', 'routing', 'services',
      'startup', 'vrrp-events'
    Do you want to enter other SNMP trap categories (y|n)? y
      Please enter SNMP trap categories: vrrp-events
  Do you want to configure Syslog server [n]: y
Syslog host address? 167.37.0.1
           port number [123]:
    Sylog facility 'all' [n]:
    Syslog facilities:
      'authorization', 'change-log',        'conflict-log', 'daemon',
      'dfc',           'explicit-priority', 'external',     'firewall',
      'ftp',           'interactive-commands', 'kernel',    'log-prefix',
      'ntp',           'pfe',               'security',     'user'
    Syslog severities:
```

```
      'alert', 'any',  'criticial', 'emergency', 'error',
      'info',  'none', 'notice',    'warning'
    Please enter syslog facility name: change-log
    Please enter severity: warning
    Do you want to enter more syslog facilities [n]?
  Do you want to configure NTP [n]: y
    NTP server address: 168.37.0.1


JNU controller configuration completed
```

jnu-commit

Syntax

```
op
  jnu-commit
```

Release Information Command introduced in JNU 1.0.

Description Commits the configuration on the controller and on the satellite devices.

jnu-order-satellites

Syntax

```
op
  jnu-order-satellites
  insert device-name-1 (before | after ) device-name-2)
```

Release Information Command introduced in JNU 1.0.

Description Change the order of satellite devices when the software displays information about the devices.

Options

device-name-1—Name of the satellite device to be inserted.

before —(Optional) Insert *device-name-1* before *device-name-2*.

after —(Optional) Insert *device-name-1* after *device-name-2*.

device-name-2—Name of the satellite device before or after which *device-name-1* is inserted.

op

Syntax

op
<*filename*>
<jnu-commit>
<remote.op *device-id,device-id* command "*command-sent-to-satellite*">
<url *url*>

Description

Runs operational commands on the controller.

Options

filename —(Optional) Runs the specified op script in the /var/db/scripts/op directory.

jnu-commit—(Optional) Begins a commit operation on the controller and satellites.

url *url*—— (Optional) Runs the op script specified in the URL.

jnu-remote

Syntax

```
op
  jnu-remote
    command "operational-mode-command"
    device device-name
```

Release Information Command introduced in JNU 1.0.

Description Forwards the specified Junos OS operational mode commands to the satellite device, and displays the results on the controller.

Options

command "*operational-mode-command*"—Name of the operational mode command that you want to send to the satellite device.

device *device-name*—Name of the satellite device to which you want to sent the command. Enter multiple satellite devices in a comma-separated list.

Required Privilege Level view

List of Sample Output jnu-remote on page 47

Sample Output

jnu-remote The following example runs the **show chassis hardware** command on satellite devices jnu-sat-ex1 and jnu1-sat-qfx2

```
user@jnu1-ctrlr> op jnu-remote device jnu-sat-ex1,jnu1-sat-qfx2 command "show chassis
hardware"
Device: jnu-sat-ex1
---------------------------------

Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                BM0211313979      EX4200-24T
Routing Engine 0 REV 13   750-033065   BM0211313979      EX4200-24T, 8 POE
FPC 0            REV 13   750-033065   BM0211313979      EX4200-24T, 8 POE
  CPU                     BUILTIN      BUILTIN           FPC CPU
  PIC 0                   BUILTIN      BUILTIN           24x 10/100/1000 Base-T
  PIC 1          REV 05   711-026017   CH0211328603      2x 10GE SFP+
    Xcvr 0       REV 01   740-030658   AD0951A01GC       SFP+-10G-USR
Power Supply 0   REV 05   740-020957   AT0511253210      PS 320W AC
Fan Tray                                                Fan Tray
Device: jnu1-sat-qfx2
---------------------------------

Hardware inventory:
Item             Version  Part number  Serial number     Description
Chassis                                P1959             QFX3500
Routing Engine 0          BUILTIN      BUILTIN           QFX Routing Engine
FPC 0            REV 15   750-036931   P1959-C           QFX 48x10G 4x40G Switch

  CPU                     BUILTIN      BUILTIN           FPC CPU
  PIC 0                   BUILTIN      BUILTIN           48x 10G-SFP+
    Xcvr 1       REV 01   740-021308   ZT521101981       SFP+-10G-SR
    Xcvr 10      REV 01   740-013111   9057111           SFP-T
```

```
  PIC 1                 BUILTIN      BUILTIN          15x 10G-SFP+
  MGMT BRD      REV 09  750-036946   BBAR8776         QFX3500-MB
Power Supply 0  Rev 04  740-032091   VE07482          QFX PS 650W AC
Power Supply 1  Rev 04  740-032091   VE06647          QFX PS 650W AC
Fan Tray 0                                           QFX Fan Tray
Fan Tray 1                                           QFX Fan Tray
Fan Tray 2                                           QFX Fan Tray
```

jnu-rollback

Syntax

```
op
  jnu-rollback
  number <number>
```

Release Information Command introduced in JNU 1.0.

Description Return to a previously committed configuration. The software saves the last 50 committed configurations. After running the **op jnu-rollback** command, you need to run the **op jun-commit** command to activate the candidate configuration.

Options

number *number*—(Optional) Configuration to return to. The range of values is from 0 through 49. The most recently saved configuration is number 0, and the oldest saved configuration is number 49. The default is 0.

jnu-show-satellites

Syntax

```
op
  jnu-show-satellites
    <detail>
```

Release Information Command introduced in JNU 1.0.

Description Display the state of each satellite.

Options **detail**—(Optional) Tests and runs the op script on the satellite devices.

Required Privilege Level view

Output Fields Table 4 on page 50 lists the output fields for the **jnu-show-satellites** command. Output fields are listed in the approximate order in which they appear.

Table 4: jnu-show-satellites Output Fields

Field Name	Field Description	Level of Output
Satellite System	Name of the satellite device	level-of-output none
State	Status of the satellite device. • Up—Satellite device is up and connected to the controller. • Down—Satellite device is not connected to the controller.	level-of-output none

jnu-show-satellites

```
user@jnu1-ctrlr> jnu-show-satellites
Satellite-System            State
jnu1-sat-ex1                      Up
jnu1-sat-qfx2                     Up
```

jnu-show-satellites detail

```
user@jnu1-ctrlr> jnu-show-satellites detail
2012-07-15 23:10:30 OMST: reading op script input details
2012-07-15 23:10:30 OMST: testing op details
2012-07-15 23:10:30 OMST: running op script 'jnu-show-satellites.slax'
2012-07-15 23:10:30 OMST: opening op script
'/var/db/scripts/op/jnu-show-satellites.slax'
2012-07-15 23:10:30 OMST: reading op script 'jnu-show-satellites.slax'
Satellite-System            State
jnu1-sat-ex1                Up
jnu1-sat-qfx2               Up
2012-07-15 23:10:35 OMST: inspecting op output 'jnu-show-satellites.slax'
2012-07-15 23:10:35 OMST: finished op script 'jnu-show-satellites.slax'
```

jnu-show-configuration

Syntax

```
op
  jnu-show-configuration
    device device-name
    source candidate | committed
    <detail>
    <display commit-script>
```

Release Information Command introduced in JNU 1.0.

Description Display the configuration of the satellite.

Options

device *device-name*—Name of the satellite device for which you want to display the configuration.

source candidate | committed—Source of the configuration to be displayed, either the candidate configuration or the committed configuration.

detail—(Optional) Display the specified level of output.

display commit-script—(Optional) Displays the commit script for the satellite

Required Privilege Level view

List of Sample Output jnu-show-configuration on page 51

Sample Output

jnu-show-configuration

```
user@jnu1-ctrlr> op jnu-show-configuration device jnu1-sat-ex1 source committed
Device: jnu1-sat-ex1
----------------------------------

## Last commit: 2012-07-13 18:25:06 PDT by jnuadmin
version 11.4R4;
groups {
    global {
        system {
            root-authentication {
                encrypted-password "$1$ZUlES4dp$OUwWo1g7cLoV/aMWpHUnC/";
            }
    jnu-module {
        system {
    jnu1-sat-ex1 {
        apply-macro set-commands {
            configuration-set-1 "set groups jnu1-sat-ex1-freeform interfaces
xe-0/0/1 description test";
        }
    }
    jnu1-sat-ex1-freeform {
        interfaces {
            xe-0/0/1 {
                description test;
            }
        }
    }
```

```
}
apply-groups [ global re0 jnu1-sat-ex1-freeform jnu-module ];
system {
    host-name jnu1-sat-ex1;
    ports {
        console log-out-on-disconnect;
    }
    login {
        user jnuadmin {
            uid 928;
            class super-user;
            shell csh;
            authentication {
                encrypted-password "$1$a7DYfEN0$huf8FDLIshFB5v6zb2WMz0";
            }
        }
    }
    syslog {
        file messages {
            any any;
        }
    }
}
chassis {
    aggregated-devices {
        ethernet {
            device-count 63;
        }
    }
}
interfaces {
    xe-0/0/1 {
        ether-options {
            802.3ad ae0;
        }
    }
    ae0 {
        aggregated-ether-options {
            lacp {
                active;
            }
        }
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members jnu;
                }
            }
        }
    }
    vlan {
        unit 101 {
            family inet {
                address 192.168.0.3/24;
            }
        }
    }
}
event-options {
    generate-event {
```

```
            event-script-timer time-interval 300;
        }
        policy jnu-controller-connectivity {
            events event-script-timer;
            then {
                event-script monitor-connectivity-to-controller.slax {
                    arguments {
                        cntrlr-ip 192.168.0.1;
                    }
                }
                event-script monitor-controller-qfx3500.slax {
                    arguments {
                        cntrlr-ip 192.168.0.1;
                    }
                }
            }
        }
        event-script {
            file monitor-connectivity-to-controller.slax;
            file monitor-controller-qfx3500.slax;
        }
    }
    snmp {
        community rw.jnu1-sat-ex1 {
            authorization read-write;
            routing-instance jnu;
        }
        trap-options {
            source-address 192.168.0.3;
        }
        trap-group jnu {
            version v2;
            categories {
                sonet-alarms {
                    ber-defect;
                    unequipped;
                }
                otn-alarms {
                    oc-lof;
                }
            }
        }
        routing-instance-access {
            access-list {
                jnu;
            }
        }
    }
    policy-options {
        prefix-list test-jnu;
        policy-statement jnu-mgmt {
            from protocol static;
            then accept;
        }
        policy-statement reject-all {
            then reject;
        }
    }
    security {
        ssh-known-hosts {
```

```
        host 192.168.0.1 {
            ecdsa-sha2-nistp256-key AAAAE2VjZHNhLXNoYTItbmlzdHAyNTY
             AAAAIbmlzdHAyNTYAAABBBNn6OCxqurMlNxK3GO8pCfHEYMK7k6IBO4
              csNOM58yowO6u+TBk3QtRkjKNzj18Py1Mx48Ml7GcfrCOmAjoTrc=;
        }
    }
}
routing-instances {
    jnu {
        instance-type vrf;
        route-distinguisher 192.168.0.3:0;
        vrf-import reject-all;
        vrf-export reject-all;
    }
}
vlans {
    jnu {
        vlan-id 4094;
        interface {
            ae0.0;
        }
        l3-interface vlan.101;
    }
}
```

CHAPTER 8

Setting Up a Basic JNU Implementation

- Example: Setting Up a Basic JNU Implementation on page 55

Example: Setting Up a Basic JNU Implementation

This example shows how to set up a basic JNU implementation.

- Requirements on page 55
- Overview on page 56
- Configuration on page 57
- Verification on page 63

Requirements

This example uses the following hardware and software components:

- One MX Series router that acts as the controller
- One EX4200 switch that acts as a satellite device
- One QFX3500 device
- Junos OS Release 11.4R4-S1 or later
- JNU 1.0

Overview

Topology

Figure 2: Basic JNU Configuration

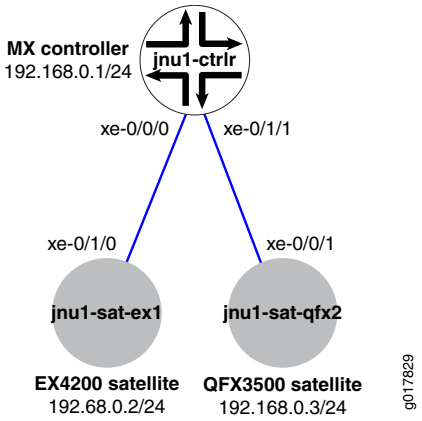


Table 5 on page 56 describes the configuration components used in this example.

Table 5: Configuration Components Used in the Basic JNU Configuration

Configuration Component	Component Name	Purpose
MX Series controller	jnu1-ctrlr	Hostname of the controller
	192.168.0.1/24	Controller management address used for communication with satellites
	xe-0/0/0	Downlink interface to jnu1-sat-ex1 satellite
	xe-0/1/1	Downlink interface to jnu1-sat-qfx2 satellite
EX4200 satellite	jnu1-sat-ex1	Hostname of the satellite
	192.168.0.2/24	Satellite management address used for communication with the controller
	xe-0/1/0	Uplink interface to the controller
QFX3500 satellite	jnu1-sat-qfx2	Hostname of the satellite
	192.168.0.3/24	Satellite management address used for communication with the controller
	xe-0/0/1	Uplink interface to the controller

NOTE: Before you begin, you must set up IP addresses on the satellite devices that are used for the connection to the controller.

Configuration

- Load the JNU Software on page 58
- Initialize the Controller on page 58
- Initialize the EX4200 Satellite on page 60
- Initialize the QFX3500 Satellite on page 60
- Commit the Configuration on page 63

Load the JNU Software

Step-by-Step Procedure

To load the JNU package onto the controller:

1. Enter the following command on the MX Series controller.

```
user@jnu1-ctrlr> request system software add jnu-1.0R1.0-signed.tgz
Installing package '/var/tmp/jnu-1.0R1-signed.tgz' ...
Verified jnu-1.0R1.tgz signed by PackageProduction_11_4_0 Adding jnu...
Available space: 556676 require: 3220
NOTICE: uncommitted changes have been saved in
/var/db/config/juniper.conf.pre-install
Mounted jnu package on /dev/md10...
Restarting bslockd ...
mgd: commit complete
Saving package file in /var/sw/pkg/jnu-1.0R1-signed.tgz ...
Saving state for rollback ...
```

Step-by-Step Procedure

To load the JNU package onto the satellite device:

1. Enter the following command on the satellite device.

```
user@jnu-satellite1> request system software add jnu-1.0R1.0-signed.tgz
Installing package '/var/tmp/jnu-1.0R1-signed.tgz' ...
Verified jnu-1.0R1.tgz signed by PackageProduction_11_4_0 Adding jnu...
Available space: 556676 require: 3220
NOTICE: uncommitted changes have been saved in
/var/db/config/juniper.conf.pre-install
Mounted jnu package on /dev/md10...
Restarting bslockd ...
mgd: commit complete
Saving package file in /var/sw/pkg/jnu-1.0R1-signed.tgz ...
Saving state for rollback ...
```

Initialize the Controller

Step-by-Step Procedure

After you install the JNU software, you need to initially configure and initialize the MX Series controller. This example configures the controller, adds two satellite devices to the controller configuration, and configures SNMP system logging, and NTP on the controller.

To initially configure the controller:

1. Enter the following command and follow the prompts. For a description of the fields used to initialize the controller, see jnu-initialize-controller.

```
user@jnu1-ctrlr> op url /var/db/scripts/op/initialize-controller.slax
Controller initializations:
  Please enter hostname [jnu-controller]:
  Please enter management IP prefix [192.168.0.1/24]:
  Please enter management VLAN id [4094]:
  Do you want to configure any satellites now [n]: y

    Please enter the number of satellites [1]: 2
    Satellite 1
      Please enter the hostname of the satellite: jnu-sat1
      Please enter the IP address of the satelltie: 192.168.0.2
      Please enter downlink interfaces to satellite: ge-0/0/0
    Satellite 2
      Please enter the hostname of the satellite: jnu-sat2
      Please enter the IP address of the satelltie: 192.168.0.3
      Please enter downlink interfaces for satellite: ge-0/0/1

  Do you want to configure SNMP [n]: y
   Do you want to enter a read-only community string (y|n)? y
      SNMP read-only community string: public
    Do you want to enter a read-only community string (y|n)? y
      SNMP read-only community string: private
    Do you want to enter SNMP trap parameters (y|n)? y
      SNMP trap target address: 169.37.0.1
    Do you want to enter SNMP trap categories (y|n)? y
    Do you want to enable SNMP trap for 'otn-alarms' (y|n)? y
      Available alarms:
        'oc-lof', 'oc-lom', 'oc-los', 'odu-ais', 'odu-bbe-threshold',
        'odu-bdi', 'odu-es-threshold', 'odu-lck', 'odu-oci',
        'odu-rx-aps-change', 'odu-sd', 'odu-ses-threshold', 'odu-sf',
        'odu-ttim', 'odu-uas-threshold', 'opu-ptm', 'otu-ais',
        'otu-bbe-threshold', 'otu-bdi', 'otu-es-threshold',
        'otu-fec-deg', 'otu-fec-exe', 'otu-iae', 'otu-sd',
        'otu-ses-threshold', 'otu-sf', 'otu-ttim', 'otu-usa-threshold',
        'wavelength-lock'
      Please enter otn-alarms: oc-lof,oc-lom
    Do you want to enable SNMP trap for 'sonet-alarms' (y|n)? y
      Available alarms:
        'ber-defect', 'ber-fault', 'line-ais',
'line-remote-defect-indication',
       'loss-of-cell', 'loss-of-frame', 'loss-of-light', 'loss-of-pointer',

        'loss-of-signal', 'path-ais', 'path-mismatch',
        'path-remote-defect-indication', 'pll-lock',
'remote-error-indication',
       'severely-errored-frame', 'unequipped', 'vt-ais', vt-label-mismatch',

        'vt-loss-of-cell', 'vt-loss-of-pointer',
'vt-remote-defect-indication',
        'vt-unequipped'
      Please enter sonet-alarms: path-ais
    Other categories:
      'authentication', 'chassis', 'configuration', 'link',
      'remote-operations', 'rmon-alarm', 'routing', 'services',
      'startup', 'vrrp-events'
    Do you want to enter other SNMP trap categories (y|n)? y
      Please enter SNMP trap categories: vrrp-events
```

```
  Do you want to configure Syslog server [n]: y
Syslog host address? 167.37.0.1
          port number [123]:
    Sylog facility 'all' [n]:
    Syslog facilities:
      'authorization', 'change-log',        'conflict-log', 'daemon',
      'dfc',           'explicit-priority', 'external',     'firewall',
      'ftp',           'interactive-commands', 'kernel',    'log-prefix',
      'ntp',           'pfe',               'security',     'user'
    Syslog severities:
      'alert', 'any',  'criticial', 'emergency', 'error',
      'info',  'none', 'notice',    'warning'
    Please enter syslog facility name: change-log
    Please enter severity: warning
    Do you want to enter more syslog facilities [n]?
  Do you want to configure NTP [n]: y
    NTP server address: 168.37.0.1


JNU controller configuration completed
```

Initialize the EX4200 Satellite

Step-by-Step Procedure

To initialize the EX4200 satellite:

1. Enter the following command on the satellite device, and follow the prompts.

```
user@jnu-satellite1> op url /var/db/scripts/op/initialize-satellite.slax
Satellite initializations:
  Please enter hostname [jnu-satellite1]: jnu1-sat-ex1

  Please enter management IP address: 10.0.0.1
  Please enter uplink IP prefix (192.168.0.2-254/24): 192.168.0.2/24
  Please enter uplink interface name: ge-0/0/0
  Please enter management VLAN id [4094]:
  Please enter controller downlink IP address [192.168.0.1]:
```

Initialize the QFX3500 Satellite

Step-by-Step Procedure

To initialize the QFX3500 satellite:

1. Enter the following command on the satellite device, and follow the prompts.

```
user@jnu-satellite> op url /var/db/scripts/op/initialize-satellite.slax
Satellite initializations:
  Please enter hostname [jnu-satellite1]: jnu1-sat-qfx2

  Please enter management IP address: 10.0.0.1
  Please enter uplink IP prefix (192.168.0.2-254/24): 192.168.0.3/24
  Please enter uplink interface name: ge-0/1/1
  Please enter uplink aggregate name [ae62]: ae0
  Please enter management VLAN id [4094]:
  Please enter controller downlink IP address [192.168.0.1]:
```

Results

The following is an example of the configuration that is initialized on the satellite device. To display this configuration from the MX Series controller, enter the following command:

```
user@jnu1-ctrlr> op jnu-show-configuration device jnu1-sat-qfx2 source candidate
```

```
    chassis {
    aggregated-devices {
      ethernet-devices {
        device-count 63;
      }
    }
}
    system {
      syslog {
        /* syslog parameters propagated from controller except source address */
        host 169.37.0.1 {
          security info;
          change-log info;
          source-address 192.168.0.3;
        }
        file messages {
          any any;
        }
      }
      ntp {
        server 132.0.1.1;
      }
    }
    snmp {
      /* Other snmp parameters propagated from controller */
      community public.hyde {
        read-only;
      }
      trap-options {
        source-address 192.168.0.3;
      }
      trap-group eas {
        version v2;
        targets {
          169.37.0.1;
        }
      }
    }
    interfaces {
      ge-0/1/1 {
        /* Using aggregate ethernet since there can be more than 1 uplink */
        gigether-options {
          802.3ad ae63;
        }
      }
      ae63 {
        /* Aggregated ethernet interface uplink connection to controller */
        aggregated-ethernet-options {
          lacp {
            active;
          }
        }
        vlan-tagging;
        unit 16385 {
          vlan-id 4094;
          family inet {
```

```
            address 192.168.0.3/24;
          }
        }
      }
    }
    policy-options {
      policy-statement jnu-management {
        /* Routes that are to be leaked from jnu-vrf to main instance */
        from {
          route-filter 169.37.0.1/32 exact;
          route-filter 132.0.1.1/32 exact;
          protocol static;
        }
        then accept;
      }
      policy-statement reject-all {
        then reject;
      }
    }
    routing-options {
      /* These configurations leak routes from jnu-vrf to main instance */
      rib-groups jnu {
        import-rib [ jnu-vrf.inet.0 inet.0 ];
        import-policy jnu-management;
      }
    }
    routing-instances {
      jnu-vrf {
        /* Routing-instance containing uplink to controller */
        instance-type vrf;
        interface ae63.16385;
        route-distinguisher 192.168.0.3:1;
        vrf-import reject-all;
        vrf-export reject-all;
        routing-options {
          routing-options {
            static {
              rib-group jnu;
              route 169.37.0.1/32 next-hop 192.168.0.1;
              route 132.0.1.1/32 next-hop 192.168.0.1;
            }
          }
        }
      }
    }
```

Commit the Configuration

Step-by-Step Procedure

JNU uses commit scripts to automate the commit process on the controller and satellite devices. You commit a configuration for the satellite devices from the controller. Each satellite device inspects the candidate configuration for errors.

You should modify and commit configuration changes only on the controller. When you commit a configuration on the controller, the flow of the commit process on the controller is as follows:

1. To commit the new configuration on the controller and on all satellite devices:

   user@jnu1-ctrlr> **op jnu-commit**

   The new configuration is validated on each satellite device.

2. If all the satellite devices successfully validate the new configuration, the controller runs the commit script and displays the following message:

```
jnu1-sat-ex1:
Configuration check succeeds
jnu1-sat-qfx2:
Configuration check succeeds
```

Verification

- Verifying That Satellite Devices Are Active on page 63

Verifying That Satellite Devices Are Active

Purpose Verify that satellite devices are active.

Action From operational mode, enter the **op jnu-show-satellites** command.

```
user@host>op jnu-show-satellites

user@controller> op jnu-show-satellites
Satellite-system            State
jnu-sat-ex1                 Up
jnu-sat-qfx2                Up
```

Meaning The display shows that the two satellites are Up.