JUNIPER NETWORKS

Junos® OS

Designing and Implementing a Junos Node Unifier Network

Release
1.2

Published: 2013-06-07

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 1.2

Revision History
January 2013—R1 Junos Node Unifier 1.2

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

Audience for Junos Node Unifier

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.

Related Documentation

- Junos Node Unifier Overview on page 4
- Basic Architecture of a JNU Network on page 5
- Terms Used in the JNU Documentation on page 5
- JNU Management Plane Overview on page 7

Junos Node Unifier Overview

Junos Node Unifier (JNU) allows you to configure and manage many Juniper Networks platforms running Junos OS from one MX Series router. You can use JNU to manage thousands of 1-Gigabit 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
- 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.

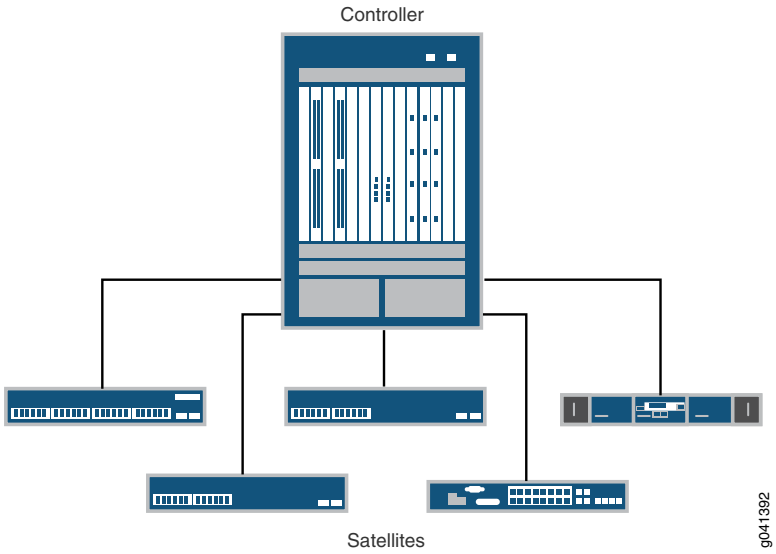
Related Documentation

- Basic Architecture of a JNU Network on page 5
- Terms Used in the JNU Documentation on page 5
- JNU Management Plane Overview on page 7
- Example: Setting Up a Basic JNU Implementation on page 71

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 Junos operating system (Junos OS), such as EX Series Ethernet switches, QFX Series devices, and ACX Series Universal Access routers.

Figure 1: Basic JNU Architecture



Related Documentation

- Junos Node Unifier Overview on page 4
- Terms Used in the JNU Documentation on page 5
- JNU Management Plane Overview on page 7
- Example: Setting Up a Basic JNU Implementation on page 71

Terms Used in the JNU Documentation

Table 1 on page 5 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	Platforms that are managed by the controller.

Related Documentation

- Junos Node Unifier Overview on page 4
- Basic Architecture of a JNU Network on page 5
- JNU Management Plane Overview on page 7

CHAPTER 2

Understanding the JNU Architecture

JNU Management Plane Overview

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

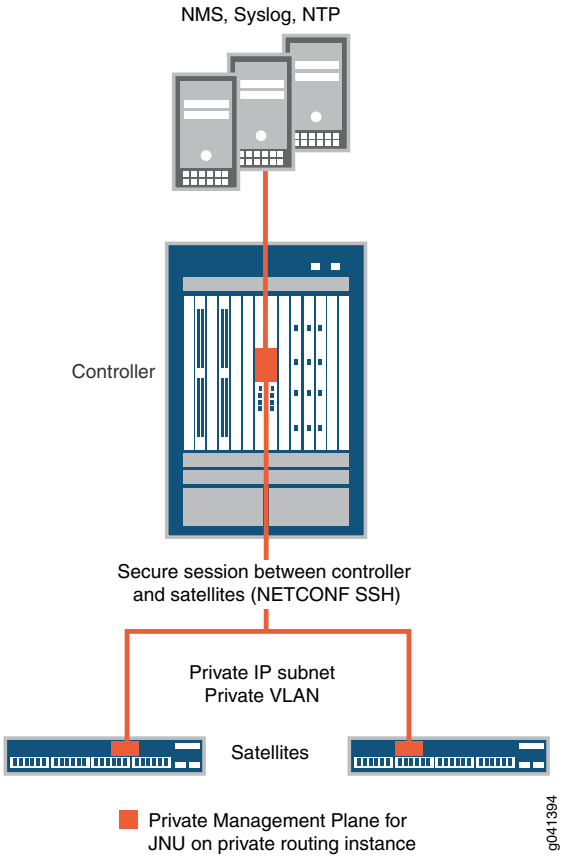
Related Documentation

- JNU Management Network on page 8
- JNU Data Plane Overview on page 10
- Junos Node Unifier Overview on page 4

JNU Management Network

The JNU architecture provides a private management plane for JNU that is separate from the control plane and the data plane. This design provides maximum performance and reliability and the ability to efficiently scale the JNU network. Figure 2 on page 8 shows a basic JNU management network. This network is created during the JNU initialization process.

Figure 2: JNU Management Network



The JNU management network is created during the JNU initialization process. The process creates a private network for in-band management, and the configuration is placed in a configuration group on the controller and on each satellite device.

The management network has the following characteristics:

- Ethernet interfaces are used for the connection between the controller and the satellites and between the controller and network management systems (NMSs). A private VLAN between the controller and the satellite devices is used to separate JNU management traffic from data plane traffic.

  During the initialization process, you specify the physical interfaces, VLAN IDs, and IP addresses to be used in the management network for the downlink connection from the controller to the satellites, and for the uplink connection from the satellites to the controller.

  By default the software places the Ethernet interfaces into an aggregated Ethernet bundle. If you specify only one physical interface during initialization, you have the option to not use aggregated Ethernet.

- The following private routing instances are created on the controller during the initialization process. These routing instances are not visible outside of the JNU Network.

  - A private VPN routing and forwarding (VRF) routing instance to provide address for Layer 3 VPNs. The VRF routing instance makes it possible to reuse the management network IP addresses in the data plane.
  - A private virtual-switch routing instance is created on the controller for the Layer 2 VPN. The virtual-switch routing instances makes it possible to reuse the VLAN IDs in the data plane.

    An integrated routing and bridging (IRB) interface is created that is used to provide IP addresses to the virtual switch.

- If supported, a routing instance is created on the satellite device that contains the uplink configuration from the satellite to the controller.
- A secure NETCONF-over-SSH connection is created between the controller and the satellite device.

Network Management in the Management Plane

During the controller initialization process, you have the option of setting up SNMP, system logging, and NTP. If you choose to set up these features, the initialization process creates a network configuration over an Ethernet interface to the NMS servers. The configuration includes static routes to these servers in the VPN routing and forwarding (VRF) routing instance on the controller.

The initialization process creates a Network Address Translation (NAT) configuration that is used to translate the source address of the traffic sent to the SNMP or syslog server so that all network management traffic from the satellite devices originates from a source address on the controller.

You do not need a license to use NAT with the JNU management plane.

Related Documentation

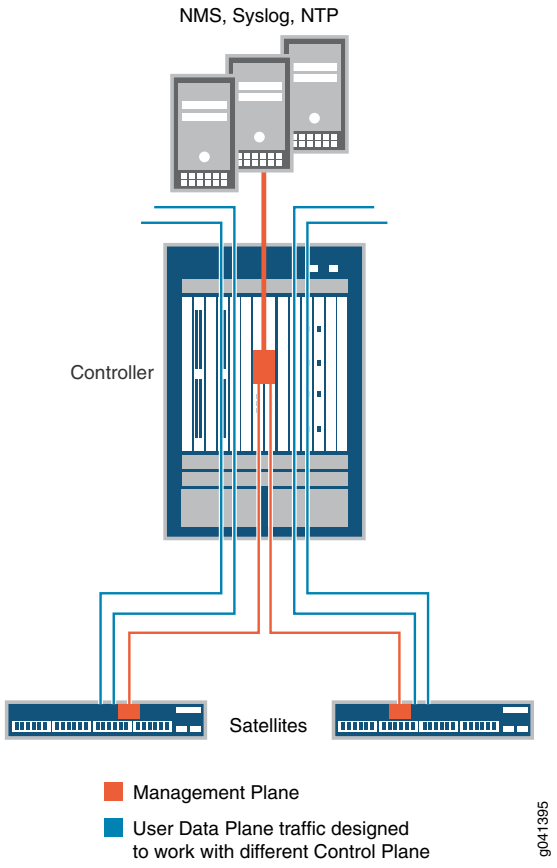
- JNU Management Plane Overview on page 7
- JNU Data Plane Overview on page 10
- Initializing JNU Mode on the Controller on page 22
- Running the Satellite Initialization Process on page 30
- Centralized Collection of SNMP Statistics and Log Messages Overview on page 85

JNU Data Plane Overview

The data plane of the controller and satellite devices, which is responsible for forwarding user data, is separate from the management plane. If your satellite device supports routing instances, the management configuration is placed in a routing instance, and you can reuse IP addresses that were used for JNU management.

For load balancing and fast recovery, you can use link aggregation of both the management interfaces and data plane instances.

Figure 3: JNU Architecture with Data Plane



Related Documentation

- JNU Management Plane Overview on page 7
- JNU Management Network on page 8

PART 2

Planning a JNU Implementation

CHAPTER 3

Planning Overview

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 link aggregation (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 MX960 router supports up to 176 10-Gigabit Ethernet interfaces. It can therefore manage up to 176 satellite devices on the 10-Gigabit Ethernet interfaces.

Related Documentation

- Supported Platforms for JNU Satellite Devices on page 15
- Junos Node Unifier Overview on page 4

Supported Platforms for JNU Satellite Devices

JNU 1.2 supports the following satellite devices:

- ACX1000 Universal Access Router
- ACX2000 Universal Access Router
- EX3200 Ethernet Switch
- EX3300 Ethernet Switch
- EX4200 Ethernet Switch
- EX4500 Ethernet Switch
- EX4550 Ethernet Switch

- EX6200 Ethernet Switch
- QFX 3500 devices

Related Documentation

- Platform Considerations for the JNU Controller on page 15
- Junos Node Unifier Overview on page 4

PART 3

Implementing JNU

CHAPTER 4

Best Practices for Configuring JNU

Naming Your JNU Controller and Satellite Devices

It is important to plan the naming of controller and satellite devices in a JNU group so that you can easily identify the satellites that belong to the same group. The hostname of satellites is used in SNMP community strings and system log prefixes to identify the satellite associated with the SNMP message or system log message.

A naming structure like the following is recommended:

- jnu1-ctrl as the controller hostname
- jnu1-sat1, jun1-sat2, jun1-sat3, and so on as the satellite hostnames

Junos OS Releases on the JNU Controller and Satellites

We recommend that you run the same Junos OS release on the controller and on the satellite devices.

CHAPTER 5

Getting Started with the JNU Software

Installing the JNU Software on the Controller

To load the JNU package onto the controller:

- Enter the following command on the MX Series controller. For example:

```
user@jnu1-ctrlr> request system software add jnu-1.2R1.2-signed.tgz
Installing package '/var/tmp/jnu-1.2R1.2-signed.tgz' ...
Verified jnu-1.2R1.2.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-1.2R1.2-signed.tgz ...
Saving state for rollback ...
```

Related Documentation

- Initializing JNU Mode on the Controller on page 22
- Installing the JNU Software on Satellite Devices on page 22

Installing the JNU Software on Satellite Devices

To load the JNU package onto the satellite device:

- Enter the following command on the satellite device. For example:

```
user@jnu-satellite1> request system software add jnu-1.2R1.2-signed.tgz
Installing package '/var/tmp/jnu-1.2R1.2-signed.tgz' ...
Verified jnu-1.2R1.2.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-1.2R1.2-signed.tgz ...
Saving state for rollback ...
```

Related Documentation

- Running the Satellite Initialization Process on page 30
- Installing the JNU Software on the Controller on page 21
- Initializing JNU Mode on the Controller on page 22

Initializing JNU Mode on the Controller

NOTE: In this guide, all the instances of JNU mode refer to both the JNU feature-rich and port-extender modes, unless explicitly stated otherwise.

After you install the JNU software, you need to initially configure and initialize the MX Series controller. The initialization process creates a JNU management plane configuration on the controller and places it in a configuration group called jnu-controller-mgmt. The management plane configuration involves interfaces, internal routing-instance, virtual-switch bridging, SNMP, system logs, NTP, and NAT in the main instance of the configuration.

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

When you initialize the controller and the satellite devices, you must be logged in to the controller or satellite as the root user. The initialization process creates a user account called jnuadmin, which the controller uses to log in to the satellites. After the initialization process is complete, log in to the controller using the jnuadmin user account.

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

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

This example configures the controller, adds two satellite devices to the controller configuration.

To initially configure the controller:

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

```
user@jnu1-ctrlr> op url /var/db/scripts/op/jnu-initialize-controller.slax
Controller initializations:
  Please select the JNU mode:
    1. Feature-rich mode
         The controller and satellite features are supported. Configurations
         are forwarded to the satellites when the target device is a
         satellite managed by this controller.
    2. Port Extender mode
  Interfaces on the satellites are extended to the satellites.
         Features are configured on the extended ports.

    JNU mode (1/2)? 2
  Please enter hostname [jnu-controller]:
  Please enter management IP address: 137.34.1.1
  Please enter JNU downlink IP prefix [192.168.0.1/24]:
  Please enter JNU 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 satellite management IP address: 10.0.0.1
      Please enter the uplink IP address of the satellite: 192.168.0.2
      Please enter downlink interfaces to satellite: ge-0/0/0
        Do you want to use aggregated-ethernet for the downlink [y]: y
        Please enter downlink aggregate name [ae31]:
    Satellite 2
      Please enter the hostname of the satellite: jnu-sat2
      Please enter satellite management IP address: 10.0.0.2
      Please enter the uplink IP address of the satellite: 192.168.0.3
      Please enter downlink interfaces for satellite: ge-0/0/1,ge-0/0/2
  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-write community string (y|n)? y
      SNMP read-write 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
    Please enter SNMP trap group name: public
    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]:
    Syslog 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: info
    Do you want to enter more syslog facilities [n]?
  Do you want to configure NTP [n]: y
    NTP server address: 188.88.0.1

JNU controller configuration completed
```

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

```
groups {
  jnu-controller-mgmt {
    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;
  }
}
interfaces {
```

```
    /* 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;
                }
              }
            }
          }
        }
      }
    }
  }
}
```

Related Documentation

- jnu-initialize-controller on page 54
- Installing the JNU Software on the Controller on page 21
- Initializing JNU Mode on Satellite Devices on page 30

Initializing JNU Mode on Satellite Devices

Running the Satellite Initialization Process

When you initialize the satellite device, the software creates a management configuration on the 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.

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. When you initialize the satellite devices, you must be logged in to the satellite as the root user.

The satellite initialization process creates a configuration as follows:

- Creates a user account called jnuadmin, which the controller uses to log in to the satellites. After the initialization process is complete, log in to the controller using the jnuadmin user account.
- Loads an SSH public key onto the satellite device and sets up a NETCONF-over-SSH connection for use between the controller and the satellite device.
- Creates a configuration group called jnu-satellite-mgmt. The configuration includes both the configuration resulting from parameters that you specify during the satellite initialization process and the configuration that is propagated from the controller.

Table 2 on page 31 describes the fields in the satellite initialization process.

Table 2: JNU Satellite Initialization Fields

Field	Description
Please select the JNU mode	JNU mode in which the satellite device in a JNU group must function. Two JNU modes available are: • **1. Feature-rich mode**—This mode is also called non-port-extender mode. In this mode, the interfaces of the satellites are not expanded on the controller; they behave as separate physical interfaces. You modify and commit configuration changes for satellite devices only on the controller and commit the configurations to the satellites. • **2. Port Extender mode**—In this mode, a JNU group that consists of the controller and a number of satellites is regarded as a single, unified network entity, with the controller owning all the interface resources, including those residing on the satellites (remote line-cards) as extended ports. A satellite interface functions as the satellite port that is extended to the controller. In this mode, when an interface that resides on the satellite is enabled within the JNU group, a service VLAN (S-VLAN) ID or tag is used to transmit the data traffic from the remote interface of the satellite to the controller. NOTE: Port-extender mode is supported in JNU Release 1.3 and later. It is necessary to select the same mode—feature-rich or port-extender— on the controller and the satellites. The JNU management plane configurations that are enabled in both the modes are identical. If you choose the mode to be different between the controller and the satellites are different, although the controller might still communicate with the satellites, the JNU operations do not work correctly. Enter **1** to enable feature-rich mode. Enter **2** to enable port-extender mode.
Please enter hostname	Hostname for the satellite. We recommend planning the names of a JNU group so that it is easy to identify which satellites and controllers belong to a group.
Please enter management IP address	Source address the satellite uses when it sends SNMP or system log messages to the controller. This address must match the satellite management IP address that you configured for the satellite during the controller initialization process.
Please enter uplink IP prefix (192.168.0.2-254/24)	IP address of the satellite used in the JNU management network for uplink connections to the controller.
Please enter uplink interface name	ID of the interface used in the JNU management network for the uplink connection to the controller. Enter multiple interfaces in a comma-separated list. The JNU software places multiple interfaces into an aggregated Ethernet bundle.
Do you want to use aggregated-ethernet for the uplink	If you enter only a single uplink interface ID, you have the option of not using aggregated Ethernet. If your satellite device does not support aggregated Ethernet, enter **n**.
Please enter uplink aggregate name	If you are using aggregated Ethernet, enter a name for the bundle.
Please enter management VLAN id [4094]	VLAN ID used on the management network.
Please enter controller downlink IP address [192.168.0.1]	IP address of the controller used on the management network for communication with the satellite.

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/jnu-initialize-satellite.slax
Satellite initializations:
  Please select the JNU mode:
    1. Feature-rich mode
         The controller and satellite features are supported. Configurations
         are forwarded to the satellites when the target device is a
         satellite managed by this controller.
    2. Port Extender mode
  Interfaces on the satellites are extended to the satellites.
         Features are configured on the extended ports.

    JNU mode (1/2)? 2
  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
   Do you want to use aggregated-ethernet for the uplink [n]: y
   Please enter uplink aggregate name [ae31] 63
   Please enter management VLAN id [4094]:
   Please enter controller downlink IP address [192.168.0.1]:
```

Sample Initial Configuration for an EX Series Switch

The following is an example of the configuration that is configured and committed on the satellite device during the initialize process.

```
groups {
  jnu-satellite-mgmt {
    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 10.0.0.1;
        }
        file messages {
          any any;
        }
      }
      ntp {
        /* Using controller mgmt IP address */
        server 192.168.0.1;
      }
      services {
        ssh;
```

```
    }
  }
  snmp {
    / * Other snmp parameters propagated from controller */
    community public:jnu-satellite1 {
      read-only;
    }
    / * Other snmp parameters propagated from controller */
    community private:jnu-satellite1 {
      read-only;
    }
    trap-options {
      source-address 10.0.0.1;
    }
    trap-group public:jnu-satellite1 {
      version v2;
      / * categories are configured by user, propagate from satellite */
      categories {
        authentication;
        routing;
      }
      targets {
        169.37.0.1;
      }
    }
  }
  interfaces {
    ge-0/0/0 {
      / * 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;
        }
      }
      unit 16385 {
        family ethernet-switching {
          port-mode trunk;
          vlan {
            members all;
          }
        }
      }
    }
    vlan {
      unit 4094 {
        family inet {
          address 192.168.0.2/24;
        }
      }
    }
```

```
    }
    vlans {
      jnu {
        vlan-id 4094;
        l3-interface vlan.4094;
      }
    }
    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 to leak routes from jnu-vrf to main instance */
      rib-groups jnu {
        import-rib [ jnu.inet.0 inet.0 ];
        import-policy jnu-management;
      }
    }
    routing-instances {
      jnu-vrf {
        / * Routing-instance containing uplink to controller */
        instance-type vrf;
        interface vlan.4094;
        route-distinguisher 192.168.0.2:1;
        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;
            }
          }
        }
      }
    }
  }
}
```

Sample Initial Configuration on an ACX Universal Access Router

The following is an example of the configuration that is configured and committed on the satellite device during the initialization process.

```
groups {
```

```
jnu-satellite-mgmt {
  system {
    syslog {
      /* syslog parameters propagated from controller except source address */
      host 169.37.0.1 {
        security info;
        change-log info;
        source-address 10.0.0.1;
      }
      file messages {
        any any;
      }
    }
    ntp {
      /* Using controller mgmt IP address */
      server 192.168.0.1;
    }
    services {
      ssh;
    }
  }
  snmp {
    /* Other snmp parameters propagated from controller */
    community public:jnu-satellite1 {
      read-only;
    }
    /* Other snmp parameters propagated from controller */
    community private:jnu-satellite1 {
      read-only;
    }
    trap-options {
      source-address 10.0.0.1;
    }
    trap-group public:jnu-satellite1 {
      version v2;
      /* categories are configured by user, propagate from satellite */
      categories {
        authentication;
        routing;
      }
      targets {
        169.37.0.1;
      }
    }
  }
  interfaces {
    ge-0/0/0 {
      /* Not using AE interface on ACX */
      vlan-tagging;
      unit 16385 {
        vlan-id 4094;
        family inet {
          address 192.168.0.2/24;
        }
      }
    }
```

```
    }
    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 to leak routes from
      jnu-vrf to main instance */
      rib-groups jnu {
        import-rib [ jnu.inet.0 inet.0 ];
        import-policy jnu-management;
      }
    }
    routing-instances {
      jnu-vrf {
        /* Routing-instance containing
        uplink to controller */
        instance-type vrf;
        interface ge-0/0/0.16385;
        route-distinguisher 192.168.0.2:1;
        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;
            }
          }
        }
      }
    }
  }
}
```

Sample Initial Configuration on an EX3300 Ethernet Switch

The following is an example of the configuration that is configured and committed an EX3300 Ethernet Switch satellite device during the initialization process. There is no routing instance configured, because the EX3300 switch does not support routing instances.

```
groups {
  jnu-satellite-mgmt {
    chassis {
      aggregated-devices {
        ethernet {
          device-count 32;
        }
      }
    }
    system {
      host-name jnu-ex3300-1;
      login {
        /* For JNU scripts to use from controller */
        user jnuadmin {
          uid 2002;
          class super-user;
          authentication {
            encrypted-password "$1$PlNup5Bb1TDx/"; ## SECRET-DATA
          }
        }
      }
      syslog {
        /* syslog parameters propagated from controller except source address */
        host 169.37.0.1 {
          security info;
          change-log info;
          source-address 10.0.0.2;
        }
        file messages {
          any any;
        }
      }
      ntp {
        /* NTP server is MX address */
        server 192.168.0.1;
      }
      services {
        ssh;
      }
    }
    snmp {
      /* Community string is propagated from controller and append with */
      /* of ":_hostname-of-satellite_", other parameters are propagated */
      /* from controller */
      community public:jnu-ex3300-1 {
        read-only;
      }
      /* Community string is propagated from controller and append with */
      /* of ":_hostname-of-satellite_", other parameters are propagated */
      /* from controller */
      community private:jnu-ex3300-1 {
        read-write;
      }
      trap-options {
        source-address 10.0.0.2;
      }
```

```
    /* trap-group name is propagated from controller appended with */
    /* ":_satellite_host_name_" */
    trap-group public:jnu-ex3300-1 {
      version v2;
      targets {
        169.37.0.2;
      }
    }
}
security {
    ssh-known-hosts {
      /* Controller downlink IP address */
      host 192.168.0.1 {
        ecdsa-sha2-nistp256-key AAAAE2VjZHNhLXNoYTItbmlzdHAyNT;
      }
    }
}
interfaces {
    ge-0/0/0 {
      /* Using aggregate ethernet since there can be more than 1 uplink */
      ether-options {
        802.3ad ae31;
      }
    }
    ae31 {
      aggregated-ether-options {
        lacp {
          active;
        }
      }
      unit 0 {
        family ethernet-switching {
          port-mode trunk;
          vlan {
            members all;
          }
        }
      }
    }
    vlan {
      unit 4094 {
        family inet {
          address 192.168.0.3/24;
        }
      }
    }
}
vlans {
    jnu {
      vlan-id 4094;
      l3-interface vlan.4094;
    }
}
routing-options {
    static {
      /* Static route for the Syslog server via controller */
```

```
            route 169.37.0.1/32 next-hop 192.168.0.1;
            /* Static route for the SNMP trap server via controller */
            route 169.37.0.2/32 next-hop 192.168.0.1;
          }
        }
        event-options {
          generate-event {
            event-script-timer time-interval 300;
          }
          policy jnu-controller-connectivity {
            events event-script-timer;
            then {
              event-script monitor-controller-qfx3500.slax {
                arguments {
                  cntrlr-ip 192.168.0.1;
                }
              }
            }
          }
          event-script {
            file monitor-controller-ex3300.slax;
          }
        }
      }
    }
```

Related Documentation

- Installing the JNU Software on Satellite Devices on page 22
- jnu-add-delete-satellites on page 52
- Initializing JNU Mode on the Controller on page 22

CHAPTER 6

Configuring Junos OS Features with JNU

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 ('set', '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 set
```

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.

Related Documentation

- config-template-name on page 51
- Configuring Junos Features with JNU Free Form on page 43
- Junos Node Unifier Overview on page 4

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.

For example, to configure an interface:

```
user@jnu1-ctrlr> op config-free-form action add device jnu1-sat-ex1 command "set
  interfaces ge-1/0/0 unit 0 vlan-id 1044 family inet address 10.10.1.1"
```

To configure routing options:

```
user@jnu1-ctrlr> op config-free-form action add device jnu1-sat-ex1 command "set
  routing-options static route 172.16.0.0 next-hop 192.168.167.254 retain no-readvertise"
```

Related Documentation

- config-free-form on page 50
- Configuring Junos Features with JNU Configuration Templates on page 41
- Junos Node Unifier Overview on page 4

CHAPTER 7

Committing Configurations

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.

Related Documentation

- Commit Process for Satellite Devices That Come Online After the Commit Process on the Controller on page 46
- Returning to a Previously Committed Junos OS Configuration on page 47

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 is as follows:

1. The satellite device comes online, and the satellite and the controller discover each 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.

Related Documentation

- jnu-commit on page 53
- Commit Process for Satellites Already Connected to the Controller on page 45
- Returning to a Previously Committed Junos OS Configuration on page 47

Returning to a Previously Committed Junos OS Configuration

To return to a configuration prior to the most recently committed one:

1. (Optional) In configuration mode on the JNU controller, display previous configurations, including the configuration number, date, and time, the name of the user who committed changes; and the method of commit.

   user@jnu-ctrlr1# **rollback ?**

   ```
   <[Enter]>            Execute this command
   0                    2012-10-01 07:55:02 PDT by jnuadmin via cli
   1                    2012-10-01 07:50:22 PDT by jnuadmin via cli
   2                    2012-10-01 07:48:00 PDT by jnuadmin via cli
   3                    2012-10-01 06:57:15 PDT by jnuadmin via junoscript
   4                    2012-10-01 05:23:55 PDT by jnuadmin via cli
.
.
.
   48                   2012-09-28 04:23:15 PDT by jnuadmin via cli
   49                   2012-09-27 06:51:52 PDT by jnuadmin via junoscript
   ```

2. Specify the configuration number, 0 through 49, to which you want to return. The most recently saved configuration is number 0 (which is the default configuration to which the system returns), and the oldest saved configuration is number 49.

   user@jnu1-ctrlr> **op** jnu-rollback **0**

   In this example, configuration 0 is now the candidate configuration.

3. In operational mode on the JNU controller, commit the new candidate configuration.

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

Related Documentation

- jnu-rollback on page 63

CHAPTER 8

JNU Operational Mode Commands

Using Operational Mode Commands with JNU Overview

Operational mode commands used with JNU are at the user@host> prompt under the **op** command. You run operational mode commands on the JNU controller.

There are two types of **op** commands for JNU.

- Commands that begin with **jnu-** are directly related to operating the JNU or to sending Junos OS operational mode commands to satellite devices.
- Commands that begin with **config-** are related to configuring Junos OS features on satellite devices.
  - The **config-free-form** command allows you to send **set** commands to satellite devices.
  - All other commands that begin with **config-** are configuration templates that you configure and send to satellite devices.

Related Documentation

- op on page 69
- Configuring Junos Features with JNU Configuration Templates on page 41
- Configuring Junos Features with JNU Free Form on page 43

config-free-form

Syntax

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

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

device *device-name*—Satellite device to which you want to add or remove 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.

"set *statement-name value*"—A **set** statement and value that are supported on the satellite device. You must enclose the **set** statement and values in quotation marks. You can enter multiple **set** statements 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.

Related Documentation

- Configuring Junos Features with JNU Free Form on page 43

config-template-name

Syntax

```
op
  config-template-name
    device device-name
    action (set | delete)
    parameter parameter-value
```

Release Information Command introduced in JNU 1.0.

Description Configure Junos OS features using templates. Each template contains parameters that correspond to a set of Junos OS configuration statements. To see a list of templates, enter **op config-?** in operational mode.

Options

template-name—Name of the configuration template that you wish to configure.

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

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

parameter *parameter-value*—Parameters and values that you configure in the template. Parameters correspond to Junos OS configuration statements. To display a list of parameters in a template, enter **op config-*template-name* ?**. For example, **op config-interface ?**.

Related Documentation

- Configuring Junos Features with JNU Configuration Templates on page 41

jnu-add-delete-satellites

Syntax

```
op
  jnu-add-delete-satellites
  action set | delete
    ae-id id-value
    downlink interface-id
    ieee-802.3ad disable
    inet ip-address
    sat-mgmt-ip ip-address
    satellite satellite-name
```

Release Information Command introduced in JNU 1.0.

Description Create and configure satellite devices in the controller configuration or remove satellite devices from the configuration.

Options

action set | delete—Use **set** to create and configure a satellite device. Use **delete** to remove a satellite device configuration.

ae-id *id-value*—ID of the aggregated Ethernet interface in the range 0–479.

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

802.3ad disable—Disable the use of an aggregated ethernet interface to bundle the downlink interfaces to the satellite.

inet *ip-address*—IP address that the JNU controller uses to communicate with the satellite device.

sat-mgmt-ip *ip-address*—IP address that the satellite uses to send syslog and SNMP messages.

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

Related Documentation

- Using Operational Mode Commands with JNU Overview on page 49

jnu-commit

Syntax

```
op
  jnu-commit
```

Release Information Command introduced in JNU 1.0.

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

Related Documentation

- Commit Process for Satellites Already Connected to the Controller on page 45
- Commit Process for Satellite Devices That Come Online After the Commit Process on the Controller on page 46
- Returning to a Previously Committed Junos OS Configuration on page 47

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.

Related Documentation

- Initializing JNU Mode on the Controller on page 22
- Example: Setting Up a Basic JNU Implementation on page 71

Output Fields Table 3 on page 54 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 select the JNU mode	JNU mode in which the controller device in a JNU group must function. Two JNU modes available are: • **1. Feature-rich mode**—This mode is also called non-port-extender mode. In this mode, the interfaces of the satellites are not expanded on the controller; they behave as separate physical interfaces. You modify and commit configuration changes for satellite devices only on the controller and commit the configurations to the satellites. • **2. Port Extender mode**—In this mode, a JNU group that consists of the controller and a number of satellites is regarded as a single, unified network entity, with the controller owning all the interface resources, including those residing on the satellites (remote line-cards) as extended ports. A satellite interface functions as the satellite port that is extended to the controller. In this mode, when an interface that resides on the satellite is enabled within the JNU group, a service VLAN (S-VLAN) ID or tag is used to transmit the data traffic from the remote interface of the satellite to the controller. NOTE: Port-extender mode is supported in JNU Release 1.3 and later. It is necessary to select the same mode—feature-rich or port-extender— on the controller and the satellites. The JNU management plane configurations that are enabled in both the modes are identical. If you choose the mode to be different between the controller and the satellites are different, although the controller might still communicate with the satellites, the JNU operations do not work correctly. Enter **1** to enable feature-rich mode. Enter **2** to enable port-extender mode.
Please enter hostname	Hostname for the controller.

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

Field	Description
Please enter management IP address	Source address used when the controller sends SNMP messages or system logs to a network management system (NMS). When the controller receives SNMP messages or system logs from the satellites, it uses NAT to convert the satellite management address to this address. Therefore, all network management traffic in the JNU group uses the same source address. In addition, the controller uses this address as the source address when it sends its own network management traffic to the NMS.
Please enter JNU downlink prefix [192.168.0.1/24]	IP address of the controller used in the JNU management network for downlink connections to satellite devices.
Please enter JNU 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 satellite management IP address	IP address for the satellite used as the source address for SNMP or system log messages. This address must match the satellite management IP address that you configured for the satellite in the satellite initialization process.
Please enter the uplink IP address of the satellite	IP address of the satellite used in the JNU management network for the uplink connection to the controller.
Please enter downlink interfaces to satellite	ID of Interface used in the JNU management network for the downlink connection to the satellite. Enter multiple interfaces in a comma-separated list. The JNU software places multiple interfaces into an aggregated Ethernet bundle.
Do you want to use aggregated-ethernet for the downlink [y]	If you enter only a single downlink interface ID, you have the option of not using aggregated Ethernet. If your satellite device does not support aggregated Ethernet, enter **n**.
Please enter downlink aggregate name [ae31] 63	If you are using aggregated Ethernet, enter a name for the bundle.
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 read-only community string for the controller, and then enter the community string. Repeat this process to enter a second community string.

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

Field	Description
Do you want to enter a read-write community string (y|n)?	Enter **y** to configure a read-write 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.
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 log server.
port number [123]:	Port number of the system log 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.

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

Field	Description
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 address: 137.34.1.1
Please enter JNU downlink IP prefix [192.168.0.1/24]:
Please enter JNU 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 satellite management IP address: 10.0.0.1
      Please enter the uplink IP address of the satellite: 192.168.0.2
      Please enter downlink interfaces to satellite: ge-0/0/0
        Do you want to use aggregated-ethernet for the downlink [y]: y
        Please enter downlink aggregate name [ae31]:
    Satellite 2
      Please enter the hostname of the satellite: jnu-sat2
      Please enter satellite management IP address: 10.0.0.2
      Please enter the uplink IP address of the satellite: 192.168.0.3
      Please enter downlink interfaces for satellite: ge-0/0/1,ge-0/0/2
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-write 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-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 in which the software displays satellite devices when you run **show** commands. If you have a large number of satellites, you may want to organize them by platform or by other sort criteria.

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.

Related Documentation

- Using Operational Mode Commands with JNU Overview on page 49

jnu-remote

Syntax

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

Release Information Command introduced in JNU 1.0.

Description Forward the specified Junos OS operational mode command to the satellite device, and display 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.

Related Documentation

- Using Operational Mode Commands with JNU Overview on page 49

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.

Related Documentation

- Returning to a Previously Committed Junos OS Configuration on page 47

jnu-show-satellites

Syntax

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

Release Information Command introduced in JNU 1.0.

Description Display the state of each satellite to determine if the satellite is up and connected to the controller. You can also use this command to test and run the op script on the satellite devices.

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

Required Privilege Level view

Output Fields Table 4 on page 64 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

Related Documentation

- Running the Satellite Initialization Process on page 30

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 {
    jnu-satellite-mgmt {
      system {
        syslog {
          /* syslog parameters propagated from controller except source address */
          host 169.37.0.1 {
            security info;
            change-log info;
            source-address 10.0.0.1;
          }
          file messages {
            any any;
          }
        }
        ntp {
          /* Using controller mgmt IP address */
```

```
        server 192.168.0.1;
    }
    services {
        ssh;
    }
}
snmp {
    /* Other snmp parameters propagated from controller */
    community public:jnu-satellite1 {
        read-only;
    }
    /* Other snmp parameters propagated from controller */
    community private:jnu-satellite1 {
        read-only;
    }
    trap-options {
        source-address 10.0.0.1;
    }
    trap-group public:jnu-satellite1 {
        version v2;
        /* categories are configured by user, propagate from satellite */
        categories {
            authentication;
            routing;
        }
        targets {
            169.37.0.1;
        }
    }
}
interfaces {
    ge-0/0/0 {
        /* Not using AE interface on ACX */
        vlan-tagging;
        unit 16385 {
            vlan-id 4094;
            family inet {
                address 192.168.0.2/24;
            }
        }
    }
}
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 to leak routes from
      jnu-vrf to main instance */
      rib-groups jnu {
        import-rib [ jnu.inet.0 inet.0 ];
        import-policy jnu-management;
      }
    }
    routing-instances {
      jnu-vrf {
        /* Routing-instance containing
        uplink to controller */
        instance-type vrf;
        interface ge-0/0/0.16385;
        route-distinguisher 192.168.0.2:1;
        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;
            }
          }
        }
      }
    }
  }
}
```

op

Syntax

```
op
  config-free-form
    device device-name
    action (add | delete)
    "set statement-name value"
  config-free-form-precommit
  config-template-name
    device device-name
    action (set | delete)
    parameter-name parameter-value
  jnu-add-delete-satellites
    action set | delete
      ae-id id-value
      downlink interface-id
      ieee-802.3ad disable
      inet ip-address
      sat-mgmt-ip ip-address
      satellite satellite-name
  jnu-commit
  jnu-initialize-controller
  jnu-order-satellites
    insert device-name-1 (before | after ) device-name-2)
  jnu-remote
    command "operational-mode-command"
    device device-name
  jnu-rollback
    number <number>
  jnu-show-satellites
    <detail>
  jnu-show-configuration
    device device-name
    source candidate | committed
    <detail>
    <display commit-script>
  <script>
<url url>
```

Description Run operational commands on the controller.

Options

config-free-form-precommit—Do not run this command. It is an internal system command.

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

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

The remaining commands are described separately.

Related Documentation

- Using Operational Mode Commands with JNU Overview on page 49

CHAPTER 9

Setting Up a Basic JNU Implementation

Example: Setting Up a Basic JNU Implementation

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

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 that acts as a satellite device
- Junos OS Release 11.4R4-S1 or later
- JNU 1.0 or later

Overview

Topology

Figure 4: Basic JNU Configuration

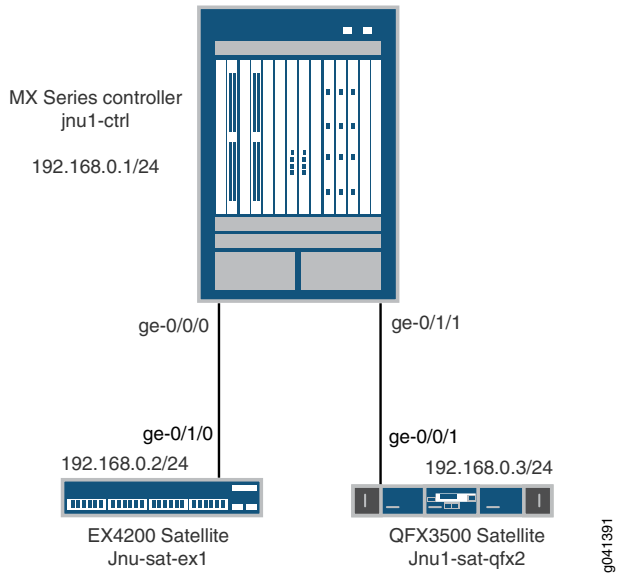


Table 5 on page 72 describes the configuration components used in this example.

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

Table 5: Configuration Components Used in the Basic JNU Configuration *(continued)*

Configuration Component	Component Name	Purpose
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

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-1.2R1.0-signed.tgz
Installing package '/var/tmp/jnu-1.2R1-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-1.2R1-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-1.2R1.0-signed.tgz
Installing package '/var/tmp/jnu-1.2R1-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-1.2R1-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/jnu-initialize-controller.slax
Controller initializations:
 Please enter hostname [jnu-controller]:
  Please enter management IP address: 137.34.1.1
  Please enter JNU downlink IP prefix [192.168.0.1/24]:
  Please enter JNU 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 satellite management IP address: 10.0.0.1
      Please enter the uplink IP address of the satellite: 192.168.0.2
      Please enter downlink interfaces to satellite: ge-0/0/0
        Do you want to use aggregated-ethernet for the downlink [y]: y
        Please enter downlink aggregate name [ae31]:
    Satellite 2
      Please enter the hostname of the satellite: jnu-sat2
      Please enter satellite management IP address: 10.0.0.2
      Please enter the uplink IP address of the satellite: 192.168.0.3
      Please enter downlink interfaces for satellite: ge-0/0/1
           Do you want to use aggregated-ethernet for the downlink [y]: y
      Please enter downlink aggregate name [ae31]:
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
```

Results The following is an example of the configuration that is initialized on the controller.

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/jnu-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
  Do you want to use aggregated-ethernet for the uplink [n]: y
  Please enter uplink aggregate name [ae31] 63
  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 configured and committed on the satellite device during the initialize process.

```
groups {
  jnu-satellite-mgmt {
    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 10.0.0.1;
      }
      file messages {
        any any;
      }
    }
    ntp {
      / * Using controller mgmt IP address */
      server 192.168.0.1;
    }
    services {
      ssh;
    }
  }
  snmp {
    / * Other snmp parameters propagated from controller */
    community public:jnu-satellite1 {
      read-only;
    }
    / * Other snmp parameters propagated from controller */
    community private:jnu-satellite1 {
      read-only;
    }
    trap-options {
      source-address 10.0.0.1;
    }
    trap-group public:jnu-satellite1 {
      version v2;
      / * categories are configured by user, propagate from satellite */
      categories {
        authentication;
        routing;
      }
      targets {
        169.37.0.1;
      }
    }
  }
  interfaces {
    ge-0/0/0 {
      / * 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;
        }
      }
      unit 16385 {
        family ethernet-switching {
          port-mode trunk;
          vlan {
            members all;
          }
        }
      }
    }
    vlan {
      unit 4094 {
        family inet {
          address 192.168.0.2/24;
        }
      }
    }
}
vlans {
    jnu {
      vlan-id 4094;
      l3-interface vlan.4094;
    }
}
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 to leak routes from jnu-vrf to main instance */
    rib-groups jnu {
      import-rib [ jnu.inet.0 inet.0 ];
      import-policy jnu-management;
    }
}
routing-instances {
    jnu-vrf {
      / * Routing-instance containing uplink to controller */
      instance-type vrf;
      interface vlan.4094;
      route-distinguisher 192.168.0.2:1;
      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;
            }
          }
        }
      }
    }
  }
```

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/jnu-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.

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 81

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.

Related Documentation

- Junos Node Unifier Overview on page 4
- Installing the JNU Software on the Controller on page 21
- Installing the JNU Software on Satellite Devices on page 22
- Initializing JNU Mode on the Controller on page 22
- Running the Satellite Initialization Process on page 30
- Commit Process for Satellites Already Connected to the Controller on page 45

PART 4

Monitoring and Troubleshooting in the JNU Network

CHAPTER 10

Monitoring in the JNU Network

Centralized Collection of SNMP Statistics and Log Messages Overview

The JNU software provides a single point of collecting SNMP statistics and sending SNMP traps to the SNMP server. You use the Junos OS SNMP proxy feature to set up the controller as a proxy SNMP agent through which the network management system (NMS) can query satellite devices.

The NMS polls the controller for SNMP statistics on both 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 process 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.

SNMP Community Strings

Each satellite in the JNU group uses a different community string that is based on the hostname assigned to the satellite during the initialization process. The format of the string is *controller-community-string.satellite-host-name*. For example, if you configure the read-only community string to be public, and the satellite hostname to be sat1, the community string for the satellite is public:sat1.

Collecting Log Messages

The JNU software provides a single point of collecting logging messages and sending them to the system log server. The controller sends all system log messages for the JNU group of satellites and controller to the server. The NAT configuration set up by the controller initialization process is used to translate the source address of the satellite devices to the source address of the controller for traffic sent to the system log server.

This process means that all logging traffic originates from one source address on the controller.

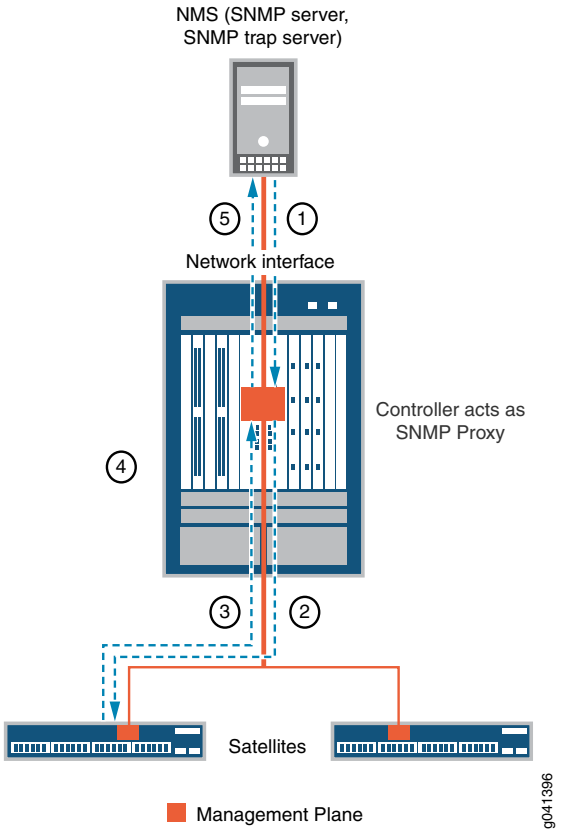
Related Documentation

- Configuring the JNU Controller as an SNMP Proxy Agent on page 91
- SNMP Get Process in the JNU Network on page 87
- SNMP Trap Process in the JNU Network on page 88
- System Logging in the JNU Network on page 89

SNMP Get Process in the JNU Network

Figure 5 on page 87 shows the SNMP Get process in the JNU network.

Figure 5: SNMP Get Process



1—SNMP server sends a Get request message to the controller with the community string for a satellite.	4—The controller receives the Get reply and processes it as an SNMP client.
2—The controller, acting as SNMP proxy, terminates the Get request from the SNMP server. It then acts as an SNMP client, and sends the request to the appropriate satellite.	5—Acting as an SNMP server, the controller then sends the Get reply message to the SNMP server.
3—The satellite sends a Get reply message to the controller.	

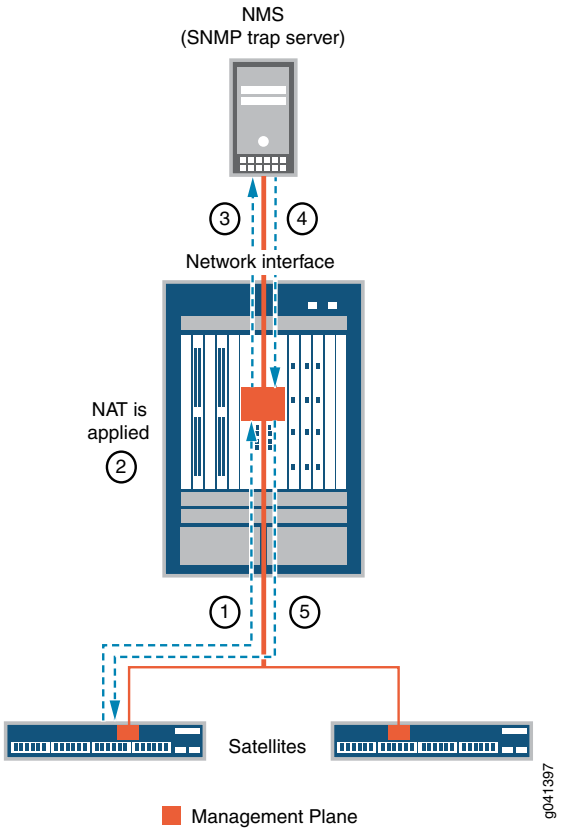
Related Documentation

- Centralized Collection of SNMP Statistics and Log Messages Overview on page 85
- Configuring the JNU Controller as an SNMP Proxy Agent on page 91
- SNMP Trap Process in the JNU Network on page 88

SNMP Trap Process in the JNU Network

Figure 6 on page 88 shows the SNMP trap process in the JNU network.

Figure 6: SNMP Trap Process



1—The satellite sends an SNMP trap notification or inform request to the controller.	4—For SNMPv3, the server responds to the inform request with an acknowledgment that it sends to the controller.
2—The controller applies NAT to the trap message so that the message source is the management IP address of the controller.	5—The controller sends the acknowledgment to the satellite.
3—The controller sends the trap message to the SNMP server.	

Related Documentation

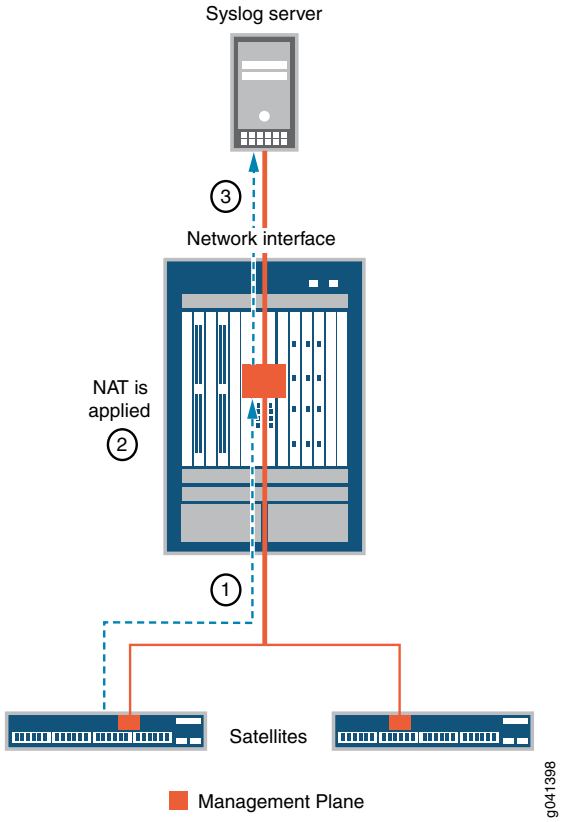
- Centralized Collection of SNMP Statistics and Log Messages Overview on page 85
- Configuring the JNU Controller as an SNMP Proxy Agent on page 91
- SNMP Get Process in the JNU Network on page 87
- System Logging in the JNU Network on page 89

System Logging in the JNU Network

Each device in the JNU group, including the controller, originates its own system log messages (called syslog messages). The hostname in messages sent from satellites and the controller is the hostname configured during the JNU initialization process. The syslog server uses the hostname to identify the JNU device that originated the message.

Figure 7 on page 89 shows how log messages are processed for JNU satellites.

Figure 7: System Logging Collection Process



1—The satellite sends a syslog message to the controller.	3—The controller sends the log message to the syslog server.
2—The controller applies NAT to the syslog message so that the message source is the management IP address of the controller.	

Related Documentation

- Centralized Collection of SNMP Statistics and Log Messages Overview on page 85
- SNMP Get Process in the JNU Network on page 87
- SNMP Trap Process in the JNU Network on page 88

Network Time Protocol (NTP) in the JNU Network

When you initialize the controller, you can specify the address of an NTP server. If you do so, the controller synchronizes its time to the external NTP server, and the controller then acts as the NTP server for all the satellites in the JNU group. During satellite initialization, JNU creates an NTP configuration that sets the NTP server address to the controller downlink IP address.

Configuring the JNU Controller as an SNMP Proxy Agent

You use the Junos OS SNMP proxy feature 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 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.

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

If you configure SNMP when you run the controller initialization process, the JNU software creates an SNMP proxy configuration with SNMPv2 support. For example:

```
proxy jnu-sat1 {
    device-name 192.168.0.2;
    version-v2c {
        snmp-community public:jnu-sat1;
    }
    routing-instance jnu-vrf;
}
```

Use this procedure if you need to add SNMPv3 support.

To configure the controller 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 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.

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 and security, and context information.

Related Documentation

- Centralized Collection of SNMP Statistics and Log Messages Overview on page 85
- SNMP Get Process in the JNU Network on page 87
- SNMP Trap Process in the JNU Network on page 88