Chapter 17

Use MIBs

This chapter describes how to determine which Management Information Bases (MIBs) are supported by a Juniper Networks release, and how to query enterprise-specific and standard MIBs to retrieve management information for the various hardware and software components of a Juniper Networks router. (See Table 39.)

Table 39: Checklist for Using MIBs

Use MIBs Tasks	Command or Action
Determine Which MIBs Are Supported by a Juniper Release on page 206	http://www.juniper.net/techpubs/software/index.html
Run Snmpwalk from an NMS System to a Juniper Router on page 207	snmpwalk [*common arguments*] *hostname community object-id*
Use SNMP Trace Operations to Monitor a Router on page 209	
1. Configure Trace Operations for SNMP on page 209	[edit] edit snmp set traceoptions flag pdu commit and-quit
2. Query a MIB With SNMPGet on page 210	snmpget *hostname community oid*
3. Display the Output for SNMP Trace Operations on page 211	show log snmp
Monitor Memory Usage on a Router on page 212	
1. Check Memory Utilization on Chassis Components on page 212	snmpwalk [*common arguments*] *hostname community object-id* show chassis routing-engine
2. Check Memory Utilization per Process on page 215	snmpwalk [*common arguments*] *hostname community object-id*
Monitor CPU Utilization on page 218	
1. Check CPU Utilization on page 218	snmpwalk [*common arguments*] *hostname community object-id*
2. Check CPU Utilization per Process on page 220	snmpwalk [*common arguments*] *hostname community object-id*
Retrieve Version Information about Router Software Components on page 223	snmpwalk [*common arguments*] *hostname community object-id*

Determine Which MIBs Are Supported by a Juniper Release

Purpose When you update the router software, you might also want to update the corresponding MIBs. Links to the MIBs related to a release are located in the *JUNOS Internet Software Network Management Configuration Guide*. This guide lists the Juniper-specific enterprise MIBs, and provides a link to Simple Network Management Protocol (SNMP) standards that list the standard MIBs supported by the JUNOS software.

In addition, a tar file that contains all the Juniper Networks enterprise-specific MIBs is included in the JUNOS software package for each release.

Action To determine MIBs supported by a Juniper release, follow these steps;

1. Enter the following URL into the address line of your browser:

   http://www.juniper.net/techpubs/software/index.html

2. Select the release you are interested in.

3. From JUNOS Configuration Guides, select Network Management.

4. From the table of contents, select JUNOS Networks Enterprise-Specific MIBs.

5. From the table of contents, select SNMP Overview.

6. From SNMP Overview, select SNMP Standards.

Run Snmpwalk from an NMS System to a Juniper Router

Purpose Snmpwalk is an SNMP application that you can use to query a MIB for information about the functioning of a router in your network. Snmpwalk uses GetNext requests to retrieve the specified information. Object identifiers (OIDs) are used to query the MIB. If the OID argument is not present, Snmpwalk searches MIB-2.

Action To run Snmpwalk for a specific OID, from a management station that has access to the router, and using a tool such as Snmpwalk, enter the following command:

user-nms# **snmpwalk** [*common arguments*] *hostname community object-id*

Sample Output

```
user-nms % snmpwalk -Os -M /volume/~/mibs -m all tp1 public .1.3.6.1.2.1.4
ipForwarding.0 = forwarding(1)
ipDefaultTTL.0 = 64
ipInReceives.0 = Counter32: 9262713
ipInHdrErrors.0 = Counter32: 0
ipInAddrErrors.0 = Counter32: 0
ipForwDatagrams.0 = Counter32: 614171
ipInUnknownProtos.0 = Counter32: 0
ipInDiscards.0 = Counter32: 0
ipInDelivers.0 = Counter32: 8648408
ipOutRequests.0 = Counter32: 1226483
ipOutDiscards.0 = Counter32: 0
ipOutNoRoutes.0 = Counter32: 0
ipReasmTimeout.0 = 60
ipReasmReqds.0 = Counter32: 0
ipReasmOKs.0 = Counter32: 0
ipReasmFails.0 = Counter32: 0
ipFragOKs.0 = Counter32: 0
ipFragFails.0 = Counter32: 0
ipFragCreates.0 = Counter32: 0
ipAdEntAddr.10.0.0.1 = IpAddress: 10.0.0.1
ipAdEntAddr.10.1.12.1 = IpAddress: 10.1.12.1
ipAdEntAddr.10.1.15.1 = IpAddress: 10.1.15.1
ipAdEntAddr.10.168.70.143 = IpAddress: 10.168.70.143
[...Output truncated...]
```

What It Means The sample output shows that the user is on a network management station (user-nms %) that has access to the router, tp1. In the command, the following options are used:

- Os—Deletes all but the last symbolic part of the OID sysUpTime.0. For example, `Timeticks: (14096763) 1 day, 15:09:27.63`.
- -M—Compiles the MIB and gives a path or location to the MIBs.
- -m—Uses the files in the directory pointed to by the -M option.
- all—Uses all the files in the directory pointed to by the -M option.

In addition, the command includes the hostname tp1, the community string public, and the OID .1.3.6.1.2.1.4.

The OID in this example is from RFC 2096, *IP Forwarding Table MIB*, which displays multipath IP routes that have the same network number but different network masks.

Before you can retrieve SNMP information from a router, you must have the minimum SNMP configuration for that router. Following is the minimum SNMP configuration required:

```
[edit]
snmp {
     community public {
         authorization read-only;
     }
 }
```

With this configuration, the system responds to SNMP Get, GetNext, and GetBulk commands that contain the community string public.

For more detailed information on configuring SNMP on a router, see the *JUNOS Network Management Configuration Guide*.

Use SNMP Trace Operations to Monitor a Router

Purpose Tracing operations record more detailed messages about the operation of SNMP, such as the various types of routing protocol packets sent and received, and routing policy actions. In this section, traceoptions are configured on a router, a MIB object is queried through a network management station, and the action of the query is verified with a log file on the router.

NOTE: Traceoptions, in general, requires extra router resources. It is recommended that you do not leave it on permanently.

Steps To Take To use SNMP traceoptions to monitor a router, follow these steps:

1. Configure Trace Operations for SNMP on page 209
2. Query a MIB With SNMPGet on page 210
3. Display the Output for SNMP Trace Operations on page 211

Step 1: Configure Trace Operations for SNMP

Purpose Define tracing for SNMP to access more granular information about the packets sent and received through SNMP.

Action To configure SNMP tracing operations, follow these steps:

1. In configuration mode, go to the following hierarchy level:

   ```
   [edit]
   user@R1# edit snmp
   ```

2. Configure trace operations:

   ```
   [edit snmp]
   user@R1# set traceoptions flag pdu
   ```

3. Commit the configuration:

   ```
   user@R1# commit and-quit
   commit complete
   Exiting configuration mode
   ```

Sample Output

```
user@R1> show configuration snmp
view all {
    oid .1 include;
}
view system {
    oid system;
}
community public {
    view all;
    authorization read-only;
}
```

```
community private {
    view system;
    authorization read-write;
}
traceoptions {
    flag pdu;
}
```

What It Means The sample output shows a configuration for SNMP that includes traceoptions. The pdu flag is configured, which results in the generation of SNMP request and response packets. The output for the tracing operation is placed into various log files in the /var/log directory.

Protocol-specific tracing operations override any equivalent operations that you specify in the global traceoptions statement. If there are no equivalent operations, they supplement the global tracing options. If you do not specify any protocol-specific tracing, the routing protocol inherits all the global tracing operations.

Step 2: Query a MIB With SNMPGet

Purpose Send an SNMP request to check that the SNMP configuration is correct.

Action To query a MIB with SNMPGet, enter the following command:

user@nms % **snmpget** *hostname community oid*

Sample Output

```
user-nms % snmpget tp1 public .1.3.6.1.2.1.1.1.0
system.sysDescr.0 = m7i internet router, kernel 6.0R1.5

user-nms % snmpget tp1 public sysDescr.0
system.sysDescr.0 = m7i internet router, kernel 6.0R1.5
```

What It Means The sample output shows a query from a network management station (nms) for the description of the system running on the router tp1. The OID is entered in numerical form in the command line, and a description (sysDescr.0) is obtained in the output. You can also use sysDescr.0 in the command line to obtain the same output.

Step 3: Display the Output for SNMP Trace Operations

Purpose The SNMP log file shows more granular information about the packets sent and received through SNMP. In this step, the contents of the SNMP log file snmpd are displayed to verify that both the GetRequest and the GetResponse packets appear in the output.

Action To display the output for trace operations, enter the following JUNOS command-line interface operational mode command:

```
user@host> show log snmpd
```

Sample Output

```
user@R1> show log snmpd
[...Output truncated...}
Jan 21 16:32:32 snmpd[28b6fdb3] >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Jan 21 16:32:32 snmpd[28b6fdb3] >>> Get-Request
Jan 21 16:32:32 snmpd[28b6fdb3] >>>  Source:      172.17.28.131
Jan 21 16:32:32 snmpd[28b6fdb3] >>>  Destination: 10.168.70.143
Jan 21 16:32:32 snmpd[28b6fdb3] >>>  Version: SNMPv1
Jan 21 16:32:32 snmpd[28b6fdb3] >>>  Request_id:  0x28b6fdb3
Jan 21 16:32:32 snmpd[28b6fdb3] >>>  Community:   public
Jan 21 16:32:32 snmpd[28b6fdb3] >>>  Error:     status=0 / vb_index=0
Jan 21 16:32:33 snmpd[28b6fdb3] >>>   OID  : sysDescr.0
Jan 21 16:32:33 snmpd[28b6fdb3] >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Jan 21 16:32:33 snmpd[28b6fdb3]  <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
Jan 21 16:32:33 snmpd[28b6fdb3]  <<< Get-Response
Jan 21 16:32:33 snmpd[28b6fdb3]  <<<  Source:      10.168.70.143
Jan 21 16:32:33 snmpd[28b6fdb3]  <<<  Destination: 172.17.28.131
Jan 21 16:32:33 snmpd[28b6fdb3]  <<<  Version: SNMPv1
Jan 21 16:32:33 snmpd[28b6fdb3]  <<<  Request_id:  0x28b6fdb3
Jan 21 16:32:33 snmpd[28b6fdb3]  <<<  Community:   public
Jan 21 16:32:33 snmpd[28b6fdb3]  <<<  Error:     status=0 / vb_index=0
Jan 21 16:32:33 snmpd[28b6fdb3]  <<<   OID  : sysDescr.0
Jan 21 16:32:33 snmpd[28b6fdb3]  <<<   type : OctetString
Jan 21 16:32:33 snmpd[28b6fdb3]  <<<   value: "m7i internet router, kernel 6.0R
Jan 21 16:32:33 snmpd[28b6fdb3]  <<<          1.5"
Jan 21 16:32:33 snmpd[28b6fdb3]  <<<   HEX  : 6d 37 69 20  69 6e 74 65
Jan 21 16:32:33 snmpd[28b6fdb3]  <<<          72 6e 65 74  20 72 6f 75
Jan 21 16:32:33 snmpd[28b6fdb3]  <<<          74 65 72 2c  20 6b 65 72
Jan 21 16:32:33 snmpd[28b6fdb3]  <<<          6e 65 6c 20  36 2e 30 52
Jan 21 16:32:33 snmpd[28b6fdb3]  <<<          31 2e 35
Jan 21 16:32:33 snmpd[28b6fdb3]  <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
[...Output truncated...]
```

What It Means The sample output shows the contents of the snmpd log file, with all of the packets sent and received through SNMP. The Get-Request packet is sent from a network management station and the Get-Response packet is sent from tp1. The value in the Get-response packet is the same as that returned to the network management station in Step 2, m7i internet router, kernel 6.0R1.5.

Monitor Memory Usage on a Router

Purpose From a management station that has access to the router, you can monitor memory usage of components, applications, and associated elements that have run or are currently running on a router.

Steps To Take From a management station that has access to the router and using a tool, such as Snmpwalk, follow these steps:

1. Check Memory Utilization on Chassis Components on page 212
2. Check Memory Utilization per Process on page 215

Step 1: Check Memory Utilization on Chassis Components

Purpose The enterprise-specific chassis MIB provides information about the router and its components. Within the chassis MIB, the jnxMIBs branch contains one main subbranch, jnxBoxAnatomy, which in turn contains a section, jnxOperatingTable. Within jnxOperatingTable, you can use the jnxOperatingBuffer object to monitor memory usage on your router. (See Figure 18.)

Figure 18: Chassis MIB Tree

After each object description is a value in parenthesis, such as (1). This value can be used to enter an OID for the specific object. For example, to gather information on memory utilization, you can type the object description (jnxOperatingBuffer) or the OID (.1.3.6.1.4.1.2636.3.1.13.1.11).

Action To check memory utilization using the Juniper enterprise chassis MIB, from a management station that has access to the router, and using a tool such as Snmpwalk, enter the following commands:

user-bsd# **snmpwalk** [*common arguments*] *hostname community object-id*
user@host> **show chassis routing-engine**

Sample Output

```
user-nms % snmpwalk -Os -M /volume/~/mibs -m all tp1 public jnxOperatingBuffer
jnxOperatingBuffer.1.1.1.0 = Gauge32: 0
jnxOperatingBuffer.1.1.2.0 = Gauge32: 0
jnxOperatingBuffer.1.1.3.0 = Gauge32: 0
jnxOperatingBuffer.2.1.0.0 = Gauge32: 0
jnxOperatingBuffer.4.1.1.0 = Gauge32: 0
jnxOperatingBuffer.4.1.2.0 = Gauge32: 0
jnxOperatingBuffer.4.1.3.0 = Gauge32: 0
jnxOperatingBuffer.4.1.4.0 = Gauge32: 0
jnxOperatingBuffer.6.1.1.0 = Gauge32: 6
jnxOperatingBuffer.6.1.2.0 = Gauge32: 6
jnxOperatingBuffer.7.1.0.0 = Gauge32: 8
jnxOperatingBuffer.7.2.0.0 = Gauge32: 8
jnxOperatingBuffer.8.1.1.0 = Gauge32: 0
jnxOperatingBuffer.8.2.3.0 = Gauge32: 0
jnxOperatingBuffer.8.2.4.0 = Gauge32: 0
jnxOperatingBuffer.9.1.0.0 = Gauge32: 28
jnxOperatingBuffer.9.1.1.0 = Gauge32: 0

user-nms % snmpwalk -Os -M /volume/~/mibs -m all tp1 public jnxOperatingDescr
jnxOperatingDescr.1.1.1.0 = midplane
jnxOperatingDescr.1.1.2.0 = midplane
jnxOperatingDescr.1.1.3.0 = midplane
jnxOperatingDescr.2.1.0.0 = Power Supply A
jnxOperatingDescr.4.1.1.0 = Left Tray front fan
jnxOperatingDescr.4.1.2.0 = Left Tray second fan
jnxOperatingDescr.4.1.3.0 = Left Tray third fan
jnxOperatingDescr.4.1.4.0 = Left Tray fourth fan
jnxOperatingDescr.6.1.1.0 = CFEB Internet Processor IIv1
jnxOperatingDescr.6.1.2.0 = CFEB Internet Processor IIv1
jnxOperatingDescr.7.1.0.0 = FPC @ 0/*/*
jnxOperatingDescr.7.2.0.0 = FPC @ 1/*/*
jnxOperatingDescr.8.1.1.0 = PIC: 4x OC-3 SONET, MM @ 0/0/*
jnxOperatingDescr.8.2.3.0 = PIC: 1x Tunnel @ 1/2/*
jnxOperatingDescr.8.2.4.0 = PIC: 1x G/E, 1000 BASE-SX @ 1/3/*
jnxOperatingDescr.9.1.0.0 = Routing Engine
jnxOperatingDescr.9.1.1.0 = Routing Engine PCMCIA Card
```

```
user@R1> show chassis routing-engine
Routing Engine status:
    Temperature                 28 degrees C / 82 degrees F
    DRAM                       256 MB
    Memory utilization          28 percent
    CPU utilization:
      User                       0 percent
      Background                 0 percent
      Kernel                     6 percent
      Interrupt                  0 percent
      Idle                      94 percent
    Model                        RE-5.0
    Serial ID                    1000431687
    Start time                   2003-11-20 11:42:04 PST
    Uptime                       63 days, 2 hours, 34 minutes, 4 seconds
    Load averages:               1 minute   5 minute  15 minute
                                     0.01       0.02       0.01
```

What It Means The sample output shows the percentage of utilization for the FPC and Routing Engine. The first object, jnxOperatingBuffer, shows that the Routing Engine (9.1.0.0) has 28 percent memory utilization, the two CFEB processors are using 6 percent, and the FPCs have 8 percent memory utilization.

The second object, jnxOperatingDescr, provides a human readable description of the separate instances in the jnxOperatingBuffer object. For example, 1.1.0.0 represents the midplane, and 7.1.0.0 represents FPC @ 0/*/*.

The output for the show chassis routing-engine command shows similar information to that displayed in the output of the jnxOperatingBuffer object, with 28 percent memory utilization for the Routing Engine.

Step 2: Check Memory Utilization per Process

Purpose The standard System Application MIB (RFC 2287, *Definitions of System-Level Managed Objects for Applications*), describes a set of managed objects that are restricted to information that can be determined from the system itself. The object sysApplElmtRunMemory provides information about applications and associated elements that have run or are currently running on the host system. (See Figure 19.)

Figure 19: System Application MIB Tree

Action To check memory utilization per process, from a management station that has access to the router, and using a tool such as Snmpwalk, enter the following command:

user-bsd# **snmpwalk** [*common arguments*] *hostname community object-id*

Sample Output

```
use-nms % snmpwalk -Os -M /volume/~/mibs -m all tp1 public sysApplElmtRunMemory
sysApplElmtRunMemory.0.0.0 = Gauge32: 0 Kbytes
sysApplElmtRunMemory.0.0.2 = Gauge32: 0 Kbytes
sysApplElmtRunMemory.0.0.3 = Gauge32: 0 Kbytes
sysApplElmtRunMemory.0.0.4 = Gauge32: 0 Kbytes
sysApplElmtRunMemory.0.0.5 = Gauge32: 0 Kbytes
sysApplElmtRunMemory.0.0.6 = Gauge32: 0 Kbytes
sysApplElmtRunMemory.0.0.7 = Gauge32: 0 Kbytes
sysApplElmtRunMemory.0.0.8 = Gauge32: 0 Kbytes
sysApplElmtRunMemory.0.0.9 = Gauge32: 0 Kbytes
sysApplElmtRunMemory.0.0.10 = Gauge32: 0 Kbytes
sysApplElmtRunMemory.0.0.11 = Gauge32: 0 Kbytes
sysApplElmtRunMemory.0.0.12 = Gauge32: 0 Kbytes
sysApplElmtRunMemory.0.0.116 = Gauge32: 526164 Kbytes
sysApplElmtRunMemory.0.0.2023 = Gauge32: 416 Kbytes
sysApplElmtRunMemory.0.0.2131 = Gauge32: 1100 Kbytes
sysApplElmtRunMemory.0.0.2160 = Gauge32: 984 Kbytes
sysApplElmtRunMemory.0.0.2161 = Gauge32: 1100 Kbytes
```

```
sysApplElmtRunMemory.0.0.2174 = Gauge32: 996 Kbytes
sysApplElmtRunMemory.0.0.2324 = Gauge32: 0 Kbytes
sysApplElmtRunMemory.0.0.16781 = Gauge32: 1072 Kbytes
sysApplElmtRunMemory.0.0.18311 = Gauge32: 1284 Kbytes
sysApplElmtRunMemory.0.0.26827 = Gauge32: 1368 Kbytes
sysApplElmtRunMemory.3.1.1 = Gauge32: 4028 Kbytes
sysApplElmtRunMemory.3.2.2163 = Gauge32: 3196 Kbytes
sysApplElmtRunMemory.3.3.2185 = Gauge32: 1624 Kbytes
sysApplElmtRunMemory.3.4.2194 = Gauge32: 9768 Kbytes
sysApplElmtRunMemory.3.7.2168 = Gauge32: 2484 Kbytes
sysApplElmtRunMemory.3.9.2169 = Gauge32: 3004 Kbytes
sysApplElmtRunMemory.3.12.2172 = Gauge32: 2108 Kbytes
sysApplElmtRunMemory.3.13.2173 = Gauge32: 1888 Kbytes
sysApplElmtRunMemory.3.14.2164 = Gauge32: 1672 Kbytes
sysApplElmtRunMemory.3.15.2175 = Gauge32: 1644 Kbytes
sysApplElmtRunMemory.3.16.2165 = Gauge32: 1632 Kbytes
sysApplElmtRunMemory.3.17.2176 = Gauge32: 2716 Kbytes
sysApplElmtRunMemory.3.19.2177 = Gauge32: 1668 Kbytes
sysApplElmtRunMemory.3.20.2178 = Gauge32: 2160 Kbytes
sysApplElmtRunMemory.3.21.2179 = Gauge32: 2164 Kbytes
sysApplElmtRunMemory.3.23.2188 = Gauge32: 1688 Kbytes
sysApplElmtRunMemory.3.25.2186 = Gauge32: 1292 Kbytes
sysApplElmtRunMemory.3.26.2180 = Gauge32: 1676 Kbytes
sysApplElmtRunMemory.3.27.2181 = Gauge32: 2052 Kbytes
sysApplElmtRunMemory.3.30.2187 = Gauge32: 1236 Kbytes
sysApplElmtRunMemory.3.31.2184 = Gauge32: 1032 Kbytes
sysApplElmtRunMemory.3.34.2171 = Gauge32: 1156 Kbytes
sysApplElmtRunMemory.3.35.2047 = Gauge32: 1132 Kbytes
sysApplElmtRunMemory.3.36.2189 = Gauge32: 1836 Kbytes
sysApplElmtRunMemory.3.37.2191 = Gauge32: 1052 Kbytes
sysApplElmtRunMemory.5.5.7495 = Gauge32: 7628 Kbytes
sysApplElmtRunMemory.5.6.2167 = Gauge32: 11824 Kbytes
sysApplElmtRunMemory.5.6.26829 = Gauge32: 11880 Kbytes
sysApplElmtRunMemory.5.8.26828 = Gauge32: 7984 Kbytes
sysApplElmtRunMemory.5.28.2182 = Gauge32: 1468 Kbytes
sysApplElmtRunMemory.5.29.2183 = Gauge32: 1828 Kbytes

user-nms % snmpwalk -Os -M /volume/~/mibs -m all tp1 public sysApplElmtRunName
sysApplElmtRunName.0.0.0 = (swapper)
sysApplElmtRunName.0.0.2 = (pagedaemon)
sysApplElmtRunName.0.0.3 = (vmdaemon)
sysApplElmtRunName.0.0.4 = (bufdaemon)
sysApplElmtRunName.0.0.5 = (syncer)
sysApplElmtRunName.0.0.6 = (netdaemon)
sysApplElmtRunName.0.0.7 = (if_pfe)
sysApplElmtRunName.0.0.8 = (if_pfe_listen)
sysApplElmtRunName.0.0.9 = (cb_poll)
sysApplElmtRunName.0.0.10 = (vmuncachedaemon)
sysApplElmtRunName.0.0.11 = (scs_housekeeping)
sysApplElmtRunName.0.0.12 = (if_pic_listen)
sysApplElmtRunName.0.0.116 = mfs
sysApplElmtRunName.0.0.2023 = pccardd
sysApplElmtRunName.0.0.2131 = cron
sysApplElmtRunName.0.0.2160 = /sbin/watchdog
sysApplElmtRunName.0.0.2161 = /usr/sbin/tnetd
sysApplElmtRunName.0.0.2174 = /usr/sbin/tnp.sntpd
sysApplElmtRunName.0.0.2324 = (peer proxy)
sysApplElmtRunName.0.0.16781 = /usr/libexec/getty
sysApplElmtRunName.0.0.18311 = /usr/sbin/xntpd
sysApplElmtRunName.0.0.26827 = telnetd
sysApplElmtRunName.3.1.1 = /sbin/preinit
sysApplElmtRunName.3.2.2163 = /usr/sbin/chassisd
sysApplElmtRunName.3.3.2185 = /usr/sbin/dfwd
```

```
sysApplElmtRunName.3.4.2194 = /sbin/dcd
sysApplElmtRunName.3.7.2168 = /usr/sbin/snmpd
sysApplElmtRunName.3.9.2169 = /usr/sbin/mib2d
sysApplElmtRunName.3.12.2172 = /usr/sbin/apsd
sysApplElmtRunName.3.13.2173 = /usr/sbin/vrrpd
sysApplElmtRunName.3.14.2164 = /usr/sbin/alarmd
sysApplElmtRunName.3.15.2175 = /usr/sbin/pfed
sysApplElmtRunName.3.16.2165 = /usr/sbin/craftd
sysApplElmtRunName.3.17.2176 = /usr/sbin/sampled
sysApplElmtRunName.3.19.2177 = /usr/sbin/ilmid
sysApplElmtRunName.3.20.2178 = /usr/sbin/rmopd
sysApplElmtRunName.3.21.2179 = /usr/sbin/cosd
sysApplElmtRunName.3.23.2188 = /usr/sbin/fsad
sysApplElmtRunName.3.25.2186 = /usr/sbin/irsd
sysApplElmtRunName.3.26.2180 = /usr/sbin/nasd
sysApplElmtRunName.3.27.2181 = /usr/sbin/fud
sysApplElmtRunName.3.30.2187 = /usr/sbin/rtspd
sysApplElmtRunName.3.31.2184 = /usr/sbin/smartd
sysApplElmtRunName.3.34.2171 = /usr/sbin/inetd
sysApplElmtRunName.3.35.2047 = syslogd
sysApplElmtRunName.3.36.2189 = /usr/sbin/spd
sysApplElmtRunName.3.37.2191 = /usr/sbin/eccd
sysApplElmtRunName.5.5.7495 = /usr/sbin/rpd
sysApplElmtRunName.5.6.2167 = /usr/sbin/mgd
sysApplElmtRunName.5.6.26829 = mgd: (mgd) (user)/dev/ttyp0
sysApplElmtRunName.5.8.26828 = -cli
sysApplElmtRunName.5.28.2182 = /usr/sbin/ppmd
sysApplElmtRunName.5.29.2183 = /usr/sbin/lmpd
```

What It Means The sample output shows the total amount of real system memory, measured in kilobytes, currently allocated to the processes retrieved by the sysApplElmtRunMemory object.

The sysApplElmtRunMemory object shows granular, per-process information about memory usage. For example, the sampled process (3.17.2176) is using 2716 kilobytes of memory.

The sysApplElmtRunName object provides a description of the separate instances displayed in the sysApplElmtRunMemory object. For example, the sampled process is represented by the OID 3.17.2176.

Monitor CPU Utilization

Purpose You can monitor CPU utilization using the Juniper specific enterprise chassis MIB and the standard system application MIB (RFC 2287, *Definitions of System-Level Managed Objects for Applications*).

Steps To Take From a management station that has access to the router, and using a tool such as Snmpwalk, follow these steps:

1. Check CPU Utilization on page 218
2. Check CPU Utilization per Process on page 220

Step 1: Check CPU Utilization

Purpose The enterprise-specific chassis MIB provides information about the router and its components. Within the chassis MIB, the jnxMIBs branch contains one main subbranch, jnxBoxAnatomy, which in turn contains a section, jnxOperatingTable. Within jnxOperatingTable, and under the jnxOperatingEntry, you can use the jnxOperatingCPU object to monitor the CPU on your router. (See Figure 20.)

Figure 20: Chassis MIB Tree

After each object description is a value in parenthesis, such as (1). This value can be used to enter an OID for the specific object. For example, to gather information on the CPU, you can type the object description (jnxOperatingCPU) or the OID (.1.3.6.1.4.1.2636.3.1.13.1.8).

Action To check CPU utilization using the Juniper enterprise chassis MIB, from a management station that has access to the router, and using a tool such as Snmpwalk, enter the following command:

user-bsd# **snmpwalk** [*common arguments*] *hostname community object-id*

Sample Output

```
user-nms % snmpwalk -Os -M /volume/~/mibs -m all tp1 public jnxOperatingCPU
jnxOperatingCPU.1.1.1.0 = Gauge32: 0
jnxOperatingCPU.1.1.2.0 = Gauge32: 0
jnxOperatingCPU.1.1.3.0 = Gauge32: 0
jnxOperatingCPU.2.1.0.0 = Gauge32: 0
jnxOperatingCPU.4.1.1.0 = Gauge32: 0
jnxOperatingCPU.4.1.2.0 = Gauge32: 0
jnxOperatingCPU.4.1.3.0 = Gauge32: 0
jnxOperatingCPU.4.1.4.0 = Gauge32: 0
jnxOperatingCPU.6.1.1.0 = Gauge32: 224
jnxOperatingCPU.6.1.2.0 = Gauge32: 224
jnxOperatingCPU.7.1.0.0 = Gauge32: 2
jnxOperatingCPU.7.2.0.0 = Gauge32: 2
jnxOperatingCPU.8.1.1.0 = Gauge32: 0
jnxOperatingCPU.8.2.3.0 = Gauge32: 0
jnxOperatingCPU.8.2.4.0 = Gauge32: 0
jnxOperatingCPU.9.1.0.0 = Gauge32: 6
jnxOperatingCPU.9.1.1.0 = Gauge32: 0

user-nms % snmpwalk -Os -M /volume/~/mibs -m all tp1 public jnxOperatingDesc
jnxOperatingDescr.1.1.1.0 = midplane
jnxOperatingDescr.1.1.2.0 = midplane
jnxOperatingDescr.1.1.3.0 = midplane
jnxOperatingDescr.2.1.0.0 = Power Supply A
jnxOperatingDescr.4.1.1.0 = Left Tray front fan
jnxOperatingDescr.4.1.2.0 = Left Tray second fan
jnxOperatingDescr.4.1.3.0 = Left Tray third fan
jnxOperatingDescr.4.1.4.0 = Left Tray fourth fan
jnxOperatingDescr.6.1.1.0 = CFEB Internet Processor IIv1
jnxOperatingDescr.6.1.2.0 = CFEB Internet Processor IIv1
jnxOperatingDescr.7.1.0.0 = FPC @ 0/*/*
jnxOperatingDescr.7.2.0.0 = FPC @ 1/*/*
jnxOperatingDescr.8.1.1.0 = PIC: 4x OC-3 SONET, MM @ 0/0/*
jnxOperatingDescr.8.2.3.0 = PIC: 1x Tunnel @ 1/2/*
jnxOperatingDescr.8.2.4.0 = PIC: 1x G/E, 1000 BASE-SX @ 1/3/*
jnxOperatingDescr.9.1.0.0 = Routing Engine
jnxOperatingDescr.9.1.1.0 = Routing Engine PCMCIA Card
```

What It Means The sample output shows the percentage CPU utilization on router, tp1. The Routing Engine (9.1.0.0) has 6 percent CPU utilization, the two CFEB Internet Processors IIv1 (6.1.1.0 and 6.1.2.0) have 22 percent each, and the FPCs (7.1.0.0 and 7.2.0.0) have 2 percent each. Components with a value of zero indicate that the information is either unavailable or inapplicable.

The output for the jnxOperatingDesc object provides a description of the separate instances in the jnxOperatingCPU object. For example, 9.1.0.0 represents the Routing Engine.

Step 2: Check CPU Utilization per Process

Purpose The standard system application MIB (RFC 2287, *Definitions of System-Level Managed Objects for Applications*), describes a set of managed objects that are restricted to information that can be determined from the system itself. The object sysApplElmtRunCPU provides information about applications and associated elements that have run or are currently running on the host system. (See Figure 21.)

Figure 21: System Application MIB Tree

Action To check CPU utilization per process, from a management station that has access to the router, and using a tool such as Snmpwalk, enter the following command:

user-bsd# **snmpwalk** [*common arguments*] *hostname community object-id*

Sample Output

```
use-nms % snmpwalk -Os -M /volume/~/mibs -m all tp1 public sysApplElmtRunCPU
sysApplElmtRunCPU.0.0.0 = Timeticks: (278) 0:00:02.78
sysApplElmtRunCPU.0.0.2 = Timeticks: (487) 0:00:04.87
sysApplElmtRunCPU.0.0.3 = Timeticks: (0) 0:00:00.00
sysApplElmtRunCPU.0.0.4 = Timeticks: (1742) 0:00:17.42
sysApplElmtRunCPU.0.0.5 = Timeticks: (13899) 0:02:18.99
sysApplElmtRunCPU.0.0.6 = Timeticks: (79) 0:00:00.79
sysApplElmtRunCPU.0.0.7 = Timeticks: (0) 0:00:00.00
sysApplElmtRunCPU.0.0.8 = Timeticks: (0) 0:00:00.00
sysApplElmtRunCPU.0.0.9 = Timeticks: (0) 0:00:00.00
sysApplElmtRunCPU.0.0.10 = Timeticks: (2229) 0:00:22.29
sysApplElmtRunCPU.0.0.11 = Timeticks: (0) 0:00:00.00
sysApplElmtRunCPU.0.0.12 = Timeticks: (0) 0:00:00.00
sysApplElmtRunCPU.0.0.116 = Timeticks: (25) 0:00:00.25
sysApplElmtRunCPU.0.0.2023 = Timeticks: (0) 0:00:00.00
sysApplElmtRunCPU.0.0.2131 = Timeticks: (1103) 0:00:11.03
sysApplElmtRunCPU.0.0.2160 = Timeticks: (1599) 0:00:15.99
sysApplElmtRunCPU.0.0.2161 = Timeticks: (4) 0:00:00.04
sysApplElmtRunCPU.0.0.2174 = Timeticks: (1168) 0:00:11.68
```

```
sysApplElmtRunCPU.0.0.2324 = Timeticks: (1738) 0:00:17.38
sysApplElmtRunCPU.0.0.16781 = Timeticks: (0) 0:00:00.00
sysApplElmtRunCPU.0.0.18311 = Timeticks: (0) 0:00:00.00
sysApplElmtRunCPU.0.0.26827 = Timeticks: (2) 0:00:00.02
sysApplElmtRunCPU.3.1.1 = Timeticks: (483) 0:00:04.83
sysApplElmtRunCPU.3.2.2163 = Timeticks: (33548776) 3 days, 21:11:27.76
sysApplElmtRunCPU.3.3.2185 = Timeticks: (1314) 0:00:13.14
sysApplElmtRunCPU.3.4.2194 = Timeticks: (5282) 0:00:52.82
sysApplElmtRunCPU.3.7.2168 = Timeticks: (20380) 0:03:23.80
sysApplElmtRunCPU.3.9.2169 = Timeticks: (6703) 0:01:07.03
sysApplElmtRunCPU.3.12.2172 = Timeticks: (337) 0:00:03.37
sysApplElmtRunCPU.3.13.2173 = Timeticks: (36) 0:00:00.36
sysApplElmtRunCPU.3.14.2164 = Timeticks: (39783) 0:06:37.83
sysApplElmtRunCPU.3.15.2175 = Timeticks: (4206) 0:00:42.06
sysApplElmtRunCPU.3.16.2165 = Timeticks: (18) 0:00:00.18
sysApplElmtRunCPU.3.17.2176 = Timeticks: (61) 0:00:00.61
sysApplElmtRunCPU.3.19.2177 = Timeticks: (25) 0:00:00.25
sysApplElmtRunCPU.3.20.2178 = Timeticks: (200) 0:00:02.00
sysApplElmtRunCPU.3.21.2179 = Timeticks: (38) 0:00:00.38
sysApplElmtRunCPU.3.23.2188 = Timeticks: (3175) 0:00:31.75
sysApplElmtRunCPU.3.25.2186 = Timeticks: (44774) 0:07:27.74
sysApplElmtRunCPU.3.26.2180 = Timeticks: (17) 0:00:00.17
sysApplElmtRunCPU.3.27.2181 = Timeticks: (48950) 0:08:09.50
sysApplElmtRunCPU.3.30.2187 = Timeticks: (11) 0:00:00.11
sysApplElmtRunCPU.3.31.2184 = Timeticks: (93) 0:00:00.93
sysApplElmtRunCPU.3.34.2171 = Timeticks: (80) 0:00:00.80
sysApplElmtRunCPU.3.35.2047 = Timeticks: (1585) 0:00:15.85
sysApplElmtRunCPU.3.36.2189 = Timeticks: (30) 0:00:00.30
sysApplElmtRunCPU.3.37.2191 = Timeticks: (326) 0:00:03.26
sysApplElmtRunCPU.5.5.7495 = Timeticks: (24721) 0:04:07.21
sysApplElmtRunCPU.5.6.2167 = Timeticks: (936) 0:00:09.36
sysApplElmtRunCPU.5.6.26829 = Timeticks: (1) 0:00:00.01
sysApplElmtRunCPU.5.8.26828 = Timeticks: (25) 0:00:00.25
sysApplElmtRunCPU.5.28.2182 = Timeticks: (29234) 0:04:52.34
sysApplElmtRunCPU.5.29.2183 = Timeticks: (21) 0:00:00.21

user-nms % snmpwalk -Os -M /~/mibs -m all tp1 public sysApplElmtRunName
sysApplElmtRunName.0.0.0 = (swapper)
sysApplElmtRunName.0.0.2 = (pagedaemon)
sysApplElmtRunName.0.0.3 = (vmdaemon)
sysApplElmtRunName.0.0.4 = (bufdaemon)
sysApplElmtRunName.0.0.5 = (syncer)
sysApplElmtRunName.0.0.6 = (netdaemon)
sysApplElmtRunName.0.0.7 = (if_pfe)
sysApplElmtRunName.0.0.8 = (if_pfe_listen)
sysApplElmtRunName.0.0.9 = (cb_poll)
sysApplElmtRunName.0.0.10 = (vmuncachedaemon)
sysApplElmtRunName.0.0.11 = (scs_housekeeping)
sysApplElmtRunName.0.0.12 = (if_pic_listen)
sysApplElmtRunName.0.0.116 = mfs
sysApplElmtRunName.0.0.2023 = pccardd
sysApplElmtRunName.0.0.2131 = cron
sysApplElmtRunName.0.0.2160 = /sbin/watchdog
sysApplElmtRunName.0.0.2161 = /usr/sbin/tnetd
sysApplElmtRunName.0.0.2174 = /usr/sbin/tnp.sntpd
sysApplElmtRunName.0.0.2324 = (peer proxy)
sysApplElmtRunName.0.0.16781 = /usr/libexec/getty
sysApplElmtRunName.0.0.18311 = /usr/sbin/xntpd
sysApplElmtRunName.0.0.26827 = telnetd
sysApplElmtRunName.3.1.1 = /sbin/preinit
sysApplElmtRunName.3.2.2163 = /usr/sbin/chassisd
sysApplElmtRunName.3.3.2185 = /usr/sbin/dfwd
sysApplElmtRunName.3.4.2194 = /sbin/dcd
```

```
sysApplElmtRunName.3.7.2168 = /usr/sbin/snmpd
sysApplElmtRunName.3.9.2169 = /usr/sbin/mib2d
sysApplElmtRunName.3.12.2172 = /usr/sbin/apsd
sysApplElmtRunName.3.13.2173 = /usr/sbin/vrrpd
sysApplElmtRunName.3.14.2164 = /usr/sbin/alarmd
sysApplElmtRunName.3.15.2175 = /usr/sbin/pfed
sysApplElmtRunName.3.16.2165 = /usr/sbin/craftd
sysApplElmtRunName.3.17.2176 = /usr/sbin/sampled
sysApplElmtRunName.3.19.2177 = /usr/sbin/ilmid
sysApplElmtRunName.3.20.2178 = /usr/sbin/rmopd
sysApplElmtRunName.3.21.2179 = /usr/sbin/cosd
sysApplElmtRunName.3.23.2188 = /usr/sbin/fsad
sysApplElmtRunName.3.25.2186 = /usr/sbin/irsd
sysApplElmtRunName.3.26.2180 = /usr/sbin/nasd
sysApplElmtRunName.3.27.2181 = /usr/sbin/fud
sysApplElmtRunName.3.30.2187 = /usr/sbin/rtspd
sysApplElmtRunName.3.31.2184 = /usr/sbin/smartd
sysApplElmtRunName.3.34.2171 = /usr/sbin/inetd
sysApplElmtRunName.3.35.2047 = syslogd
sysApplElmtRunName.3.36.2189 = /usr/sbin/spd
sysApplElmtRunName.3.37.2191 = /usr/sbin/eccd
sysApplElmtRunName.5.5.7495 = /usr/sbin/rpd
sysApplElmtRunName.5.6.2167 = /usr/sbin/mgd
sysApplElmtRunName.5.6.26829 = mgd: (mgd) (user)/dev/ttyp0
sysApplElmtRunName.5.8.26828 = -cli
sysApplElmtRunName.5.28.2182 = /usr/sbin/ppmd
sysApplElmtRunName.5.29.2183 = /usr/sbin/lmpd
```

What It Means The sample output shows the number of centi-seconds of total system CPU resources consumed by a particular process. For example, the chassis process (chassisd, 3.2.2163) has consumed 3 days, or 33,548,776 centi-seconds of total system CPU resources.

The sysApplElmtRunName object retrieves the name of the OID. For example, sysApplElmtRunCPU.3.2.2163 represents the chassis process.

Retrieve Version Information about Router Software Components

Purpose RFC 2790, *Host Resources MIB,* describes a set of managed objects that are useful for managing host systems, including routers.

Action To retrieve version information about software components on the router, from a management station that has access to the router and using a tool, such as Snmpwalk, enter the following command:

user-bsd# **snmpwalk** [*common arguments*] *hostname community object-id*

Sample Output

```
user-nms % snmpwalk -Os -M /volume/~/mibs -m all tp1 public .1.3.6.1.2.1.25.6.3
hrSWInstalledIndex.2 = 2
hrSWInstalledIndex.3 = 3
hrSWInstalledIndex.4 = 4
hrSWInstalledIndex.5 = 5
hrSWInstalledIndex.6 = 6
hrSWInstalledIndex.9 = 9
hrSWInstalledName.2 = "JUNOS Base OS Software Suite [6.0R1.5]"
hrSWInstalledName.3 = "JUNOS Kernel Software Suite [6.0R1.5]"
hrSWInstalledName.4 = "JUNOS Packet Forwarding Engine Support (M7i/M10i)
[6.0R1.5]"
hrSWInstalledName.5 = "JUNOS Routing Software Suite [6.0R1.5]"
hrSWInstalledName.6 = "JUNOS Online Documentation [6.0R1.5]"
hrSWInstalledName.9 = "JUNOS Support Tools Package [6.0-20031122-unocM2]"
hrSWInstalledID.2 = OID: zeroDotZero
hrSWInstalledID.3 = OID: zeroDotZero
hrSWInstalledID.4 = OID: zeroDotZero
hrSWInstalledID.5 = OID: zeroDotZero
hrSWInstalledID.6 = OID: zeroDotZero
hrSWInstalledID.9 = OID: zeroDotZero
hrSWInstalledType.2 = operatingSystem(2)
hrSWInstalledType.3 = operatingSystem(2)
hrSWInstalledType.4 = operatingSystem(2)
hrSWInstalledType.5 = operatingSystem(2)
hrSWInstalledType.6 = application(4)
hrSWInstalledType.9 = operatingSystem(2)
hrSWInstalledDate.2 = 2003-8-10,20:34:45.0,-7:0
hrSWInstalledDate.3 = 2003-8-10,20:35:21.0,-7:0
hrSWInstalledDate.4 = 2003-8-10,20:36:30.0,-7:0
hrSWInstalledDate.5 = 2003-8-10,20:36:47.0,-7:0
hrSWInstalledDate.6 = 2003-8-10,20:36:51.0,-7:0
hrSWInstalledDate.9 = 2003-11-22,4:8:47.0,-8:0al
```

What It Means The sample output shows the version information for various software components on the router.