Using MIBs

This chapter describes the objects and MIBs that are needed to use Simple Network Management Protocol (SNMP) requests to perform the following tasks on Cisco ASR 901 and 901S routers.

Tips and Guidelines

When using SNMP to manage the Cisco ASR 901 and 901S router, be aware of the following points.

IF-MIB Caching

The Cisco ASR 901 and 901S routers implement a cache to allow continuous polling of the ifTable interface counters, without creating spikes in the CPU usage. An SNMP request for these counters returns the values that were last stored in the counter cache memory, instead of returning the current run-time value of these counters.

The ifTable counter cache is updated approximately every 10 seconds, which means that if you read the ifTable interface counters more quickly than every 10 seconds, the SNMP request might not return new values. The run-time counters do continue to increment, however, to account for the actual traffic occurring on the interfaces, and another SNMP request in 10 seconds does show the new values.

SNMP-Based and CLI-Based Counters

The SNMP specifications do not allow most SNMP-based counters to be cleared, except at system initialization. Instead, during normal operations the counters continue incrementing until they reach their maximum value, at which point they wrap around to zero and continue incrementing again.

This behavior requires the following considerations when managing the router using SNMP commands:

- 32-bit counters—A 32-bit counter wraps around to zero after reaching approximately 4.2 billion. On a busy router, this means that byte and packet counters could wrap around after only a few days. To ensure that you are maintaining the correct counts for packets and other objects, regularly poll the

desired counters and always save the previous values. Subtract the previous value from the current value, and if the difference between the two counters becomes negative, it indicates that the counters have wrapped.

To accurately total the counters over a period of several weeks or months, you might also need to keep track of the number of times that the counter wraps during this time period. You should poll the counters often enough so that they do not wrap around to zero more than once without being detected.

Tip Some SNMPv3 MIBs are beginning to include 64-bit counters, as well as 32-bit counters, for many of the same objects. If given a choice, use the 64-bit counters, because they typically will not wrap around to zero for months or years, if ever.

- Counting from a specified event or time period—SNMP-based counters begin incrementing from zero when the router is powered on, and continue incrementing until they wrap. To track the number of packets or other objects from a particular event, you must save the value of the counters at the time of the event. Then when you want to obtain a new packet count, compare the current value of the counters with the saved value.
- Comparison with command-line interpreter (CLI) values—Many **show** commands have a corresponding **clear** command that resets the counters to zero. The **clear** command, however, affects only the counters that are displayed by the CLI, not the SNMP-based counters. In addition, many CLI-based counters automatically reset whenever a certain function, such as resetting an interface, is performed. This means that the counters displayed using CLI commands are not usually the same as the counters displayed by SNMP commands. Be aware of these differences when comparing the CLI-based and SNMP-based counters.

Obtaining Basic Information About the Router

Basic information about the Cisco ASR 901 and 901S router can be obtained from objects in the following MIBs:

- OLD-CISCO-CHASSIS-MIB, page 4-2
- SNMPv2-MIB, page 4-3
- ENTITY-MIB, page 4-3

OLD-CISCO-CHASSIS-MIB

The following object in the OLD-CISCO-CHASSIS-MIB provides a convenient location to store the chassis serial number for the router, so that it can be easily retrieved when calling Cisco Technical Support:

- chassisId—Provides the serial number or ID number for the chassis, as defined by the **snmp-server chassis-id** command, which is typically used to identify the service contract and levels of service that you have purchased from Cisco Technical Support. This object defaults to the empty string, so you must use the **snmp-server chassis-id** command to set the value of this object before you can retrieve it.

```
csh% getmany -v2c 10.10.11.12 public chassisId

chassisId.0 = TBA06500113
```

SNMPv2-MIB

The following objects in the SNMPv2-MIB provide basic information about the router, its software, and other run-time information:

- sysDescr—Provides an overall description of the router, including its model number and the version of Cisco IOS software that it is running. For example:

```
csh% getmany -v2c 10.10.11.12 public sysDescr

Cisco IOS Software, 901 Software (ASR901-UNIVERSALK9-M), Version 15.1(2)SNG, RELEASE
SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2011 by Cisco Systems, Inc.
Compiled Wed 05-Oct-11 00:23 by prod_rel_team
```

- sysObjectID—Provides the specific model number, as it is defined in the CISCO-PRODUCTS-MIB. For example:

```
csh% getmany -v2c 10.10.11.12 public sysObjectId

sysObjectID.0 = ciscoProducts.ciscoASR901
```

- sysName—Provides the host name for the router, as assigned by the **hostname** command. For example:

```
csh% getmany -v2c 10.10.11.12 public sysName

sysName.0 = ASR901-Router
```

- sysUpTime—Provides the time, in hundredths of a second, since the router was last initialized. For example:

```
csh% getmany -v2c 10.10.11.12 public sysUpTime

sysUpTime.0 = 138389875
```

- sysContact—Provides the name, phone number, or other identifying information for the person or department responsible for this router, as it was entered using the **snmp-server contact** command. For example:

```
csh% getmany -v2c 10.10.11.12 public sysContact

sysContact.0 = IT Support at 408-555-1212 or epage it-support
```

- sysLocation—Provides a description of the router's location, as it was entered using the **snmp-server location** command. For example:

```
csh% getmany -v2c 10.10.11.12 public sysLocation
```

ENTITY-MIB

The following objects in the ENTITY-MIB provide basic information about the router's hardware:

- entPhysicalDescr—Provides a description of each hardware component in the router. For example, the following is a typical description for the Cisco ASR 901 chassis:

```
csh% getnext -v2c 10.10.11.12 public entPhysicalDescr

entPhysicalDescr.1 = ASR901 chassis, Hw Serial#: 65100, Hw Revision: A
```

- entPhysicalHardwareRev—Provides the hardware revision of each component, if present and supported for that particular component. For example:

```
csh% getnext -v2c 10.10.11.12 public entPhysicalHardwareRev

entPhysicalHardwareRev.1 = 1.1
```

- entPhysicalSerialNum—Provides the serial number for each component, if present and supported for that particular component. For example:

```
csh% getnext -v2c 10.10.11.12 public entPhysicalSerialNum

entPhysicalSerialNum.1 = TBC06481339
```

- entPhysicalModelName—Provides the model name for each component, if present and supported for that particular component. For example:

```
csh% getnext -v2c 10.10.11.12 public entPhysicalModelName

entPhysicalModelName.1 = ASR901
```

Tip Also see the next section for more information about the ENTITY-MIB and how to use it.

Cisco Unique Device Identifier Support

The ENTITY-MIB supports the Cisco compliance effort for a Cisco unique device identifier (UDI) standard, which is stored in IDPROM. The Cisco UDI provides a unique identity for every Cisco product. The UDI is composed of three separate data elements that must be stored in the entPhysicalTable:

- Orderable product identifier—Product Identifier (PID) is the alphanumeric identifier used by customers to order Cisco products. Two examples include NM-1FE-TX and CISCO3745. PID is limited to 18 characters and must be stored in the entPhysicalModelName object.
- Version identifier—Version Identifier (VID) is the version of the PID. The VID indicates the number of times a product has versioned in ways that are reported to a customer. For example, the product identifier NM-1FE-TX may have a VID of V04. VID is limited to three alphanumeric characters and must be stored in the entPhysicalHardwareRev object.
- Serial number—Serial number (SN) is the 11-character identifier used to identify a specific part within a product and must be stored in the entPhysicalSerialNum object. Serial number content is defined by manufacturing part number 7018060-0000.

Serial number format is defined in four fields:

  - Location (L)
  - Year (Y)
  - Workweek (W)
  - Sequential serial ID (S)

The SN label is represented as: LLLYYWWSSS.

Note The VID returns NULL for those old or existing cards with IDPROMs that do not have the VID field. Therefore, corresponding entPhysicalHardwareRev returns NULL for cards that do not have the VID field in IDPROM.

Managing Physical Components

The Cisco ASR 901 and 901S router supports a number of MIBs for the management of the router's physical components. These MIBs provide the following functions:

- Organizes the physical entities in the chassis into a containment tree that describes the relationship of each entity to all other entities
- Maps physical ports to their respective interfaces
- Provides asset information for asset tagging
- Provides firmware and software information for chassis components

See the following sections for a description of each MIB, as well as instructions on how to use the MIBs to track the components in the router:

- Managing Physical Components using ENTITY-MIB, page 4-5
- Cisco-Specific MIBS, page 4-6
- Performing Inventory Management, page 4-6

Tip To retrieve the chassis serial number for the router, retrieve the chassisId object from the OLD-CISCO-CHASSIS-MIB. This object defaults to the empty string, so you must use the **snmp-server chassis-id** command to set the value of this object before you can retrieve it.

Managing Physical Components using ENTITY-MIB

The Cisco ASR 901 and 901S routers use the ENTITY-MIB, which is defined as the standard RFC 2737, to manage its physical components, which are known as entities. An entity could be a port on a card, a slot in the chassis, or any other equipment that is installed in the router.

The ENTITY-MIB defines a set of objects that uniquely identify each entity in the router, using a hierarchical containment tree that shows how each entity relates to each other. Other MIBs can then use the objects defined by the ENTITY-MIB to provide additional information about each entity.

The following are the most important objects in the ENTITY-MIB for the management of physical entities on the router:

- entPhysicalTable—Describes each physical component (entity) in the router. The table contains a row entry for the top-most entity (the chassis) and then for each entity in the chassis. Each entry provides the name and description of the entry, its type and vendor, and a description of how the entity fits into the containment tree.
- entPhysicalIndex—Uniquely identifies each entry. This value is guaranteed to be unique across all equipment in this chassis and across all MIBs, allowing you to correlate the data from several MIBs for any particular entity.
- entAliasMappingTable—Maps each physical port's entPhysicalIndex value to the corresponding ifIndex value in the ifTable in the IF-MIB. This provides a quick way of identifying a particular port with a particular interface.
- entPhysicalContainsTable—For each physical entity, lists the entPhysicalIndex value for any child objects of the entity. This provides an easy way of creating the container tree for the router, which shows the relationship between physical entities in the chassis.

Cisco-Specific MIBS

In addition to the ENTITY-MIB, the Cisco ASR router uses the following MIBs to provide additional information about the physical components that are installed in the router:

- CISCO-ENTITY-VENDORTYPE-OID-MIB—Contains the object identifiers (OIDs) for all physical entities in the router.
- CISCO-ENVMON-MIB—Contains information about the status of environmental sensors (for voltage, temperature, fans, and power supplies). For example, this MIB reports the chassis core and inlet temperatures.

Performing Inventory Management

The ENTITY-MIB provides all of the information needed to collect an inventory of the physical components in the router.

To collect and organize the information in the ENTITY-MIB, use the following procedure.

Step 1 Collect the list of physical entities by displaying all of the entPhysicalDescr objects.

Step 2 Obtain additional information about each entPhysicalDescr object by collecting the entPhysicalVendorType, entPhysicalName, and entPhysicalClass objects. Use the index value to match the objects with their corresponding entPhysicalDescr object.

Step 3 To create the containment tree for the router, collect the EntPhysicalContainedIn object for each entPhysicalDescr object. The value in EntPhysicalContainedIn is the index number for the parent (or "container") for the corresponding entPhysicalDescr device.

Step 4 (Optional) If a parent object contains multiple children that are the same type of object, use the entPhysicalParentRelPos objects to organize the child objects into their proper order. The entPhysicalParentRelPos objects contain an integer that shows the sequential order of the child objects. This integer typically starts incrementing from 0, so that it matches the actual numbering of the physical objects (slot 0 has an entPhysicalParentRelPos value of 0, slot 1 has an entPhysicalParentRelPos value of 1, and so forth).

Note If entPhysicalParentRelPos contains –1, then the object does not have an identifiable relationship with the other objects.

Generating SNMP Traps

This section describes how to configure the Cisco ASR 901 and 901S routers to generate SNMP traps when certain events or conditions occur on the router. To use SNMP commands to configure the router to generate SNMP traps, you must define at least one target host to receive the traps, using the following procedure:

Step 1 Create an entry in the snmpTargetAddrTable, which is defined in SNMP-TARGET-MIB, for each host that is to receive traps. Each entry contains the following objects:

- snmpTargetAddrName—Unique string, up to 32 characters long, that identifies this host.
- snmpTargetAddrTDomain—The TCP/IP transport service to be used when delivering traps to this host, typically snmpUDPDomain.
- snmpTargetAddrTAddress—The transport address for the host, typically a six-octet value that is composed of the host's four-byte IP address followed by the two-byte UDP port number to which the traps should be sent.
- snmpTargetAddrTimeout—Maximum period of time, in hundredths of a second, that the Cisco ASR 901 router waits for a response from the host (if any). The default is 1500 (15 seconds).
- snmpTargetAddrRetryCount—Default number of times that the Cisco ASR 901 or 901S router resends a trap if a response is not received within the timeout period. The default value is 3 retries.
- snmpTargetAddrTagList—List of tags (defined below) that should be associated with this particular target host. If a host's tag value matches an snmpNotifyTag value, the host receives the types of notifications that are defined by the corresponding snmpNotifyType.
- snmpTargetAddrParams—Arbitrary string, up to 32 characters long, that identifies an entry in the snmpTargetParamsTable, which defines the parameters to be used in generating traps.
- snmpTargetAddrStorageType—Type of storage to be used for this row entry: volatile(2), nonVolatile(3), permanent(4), or readOnly(5). The default is nonVolatile(4).
- snmpTargetAddrRowStatus—Must be set to createAndGo(4) or createAndWait(5) to create this row entry. This object must be set only after all of the other entries in the row have been set.

Step 2 Create an entry in the snmpTargetParamsTable, which is defined in SNMP-TARGET-MIB, to define the SNMP parameters that the router should use when generating SNMP notifications. Each entry contains the following objects:

- snmpTargetParamsName—Unique string, up to 32 characters long, that defines this particular entry. This string is also used in the snmpTargetAddrParams to define the parameters to be used when sending traps to any particular host.
- snmpTargetParamsMPModel—Version of SNMP to be used in sending this trap: 0=SNMPv1, 1=SNMPv2c, and 3=SNMPv3.
- snmpTargetParamsSecurityModel—Version of SNMP security to be used in sending traps: 0=SNMPv1, 1=SNMPv2c, and 3=SNMPv3.
- snmpTargetParamsSecurityName—String, up to 32 characters long, to be used in identifying the Cisco ASR 901 or 901S router when sending traps.
- snmpTargetParamsSecurityLevel—Type of security to be used when sending traps: noAuthNoPriv(1), authNoPriv(2), and authPriv(3).
- snmpTargetParamsStorageType—Type of storage to be used for this row entry: volatile(2), nonVolatile(3), permanent(4), or readOnly(5). The default is nonVolatile(4).
- snmpTargetParamsRowStatus—Must be set to createAndGo(4) or createAndWait(5) to create this row entry. This object must be set only after all of the other entries in the row have been set.

A number of notifications and traps can also be enabled using CLI commands. Table 4-1 lists some of the most common traps, how they can be enabled through the CLI, and the situations that generate these traps.

Table 4-1 Common Notifications and Traps

Type of Trap	Configuration Command to Enable	Description
Configuration Changes	`snmp-server enable traps entity`	When ENTITY traps are enabled, the router generates an entConfigChange trap when the information in any of the following tables in the ENTITY-MIB changes: • entPhysicalTable • entAliasMappingTable • entPhysicalContainsTable **Note** The SNMP manager should also regularly poll the entLastChangeTime object to detect whether traps were missed due to throttling or transmission loss.
Environmental Changes	`snmp-server enable traps envmon`	When ENVMON traps are enabled, the router generates the following traps (defined in CISCO-ENVMON-MIB) to notify you of potential environmental problems: • ciscoEnvMonShutdownNotification—Sent when the router is about to shut down. • ciscoEnvMonTemperatureNotification—Sent when a temperature is outside its normal range. • ciscoEnvMonFanNotification—Sent when a fan fails. • ciscoEnvMonRedundantSupplyNotification—Sent when a redundant Power Entry Module fails.
Alarm is Asserted or Cleared	`snmp-server enable traps alarms`	When ALARM traps are enabled, the router generates a trap whenever an alarm is asserted or cleared for physical entities that are defined in the entPhysicalTable in the ENTITY-MIB.
SYSLOG Message is Generated	`snmp-server enable traps syslog`	By default, the Cisco ASR 901 and 901S routers log a SYSLOG message each time an alarm is asserted or cleared. To also generate a separate trap when any SYSLOG message is logged, set the clogNotificationsEnabled object to true(1). Set the clogMaxSeverity object in CISCO-SYSLOG-MIB to the maximum severity level for the SYSLOG messages that are to be stored in the CISCO-SYSLOG-MIB and for which notifications should generated. The default is 5 (warning), which indicates that SYSLOG messages of severity levels 1 through 5 are processed by the MIB.

Monitoring SYSLOG Messages

The CISCO-SYSLOG-MIB defines a number of objects that store the SYSLOG messages that the Cisco ASR 901 and 901S routers generate during its normal operations. You can regularly poll this MIB to obtain the list of SYSLOG messages that have been generated.

Message Table Objects

When enabled, SYSLOG messages are stored as an entry in the clogHistoryTable. Each clogHistoryEntry contains the following objects for each message that is stored:

- clogHistIndex—Index number that uniquely identifies each SYSLOG message that is stored in the table. This index is a 32-bit value that continually increases until it reaches its maximum value, at which point it wraps around back to 1.
- clogHistFacility—Facility identifier, up to 20 characters, of the SYSLOG message.
- clogHistSeverity—Severity level of the SYSLOG message, as defined by the SyslogSeverity textual convention, which ranges from 1 (emergency) to 8 (debug).

Note The severity numbers used in the SyslogSeverity and clogHistSeverity objects are one more than the numbers used in the actual SYSLOG messages. For example, an error SYSLOG message has a severity of 3, but SyslogSeverity uses 4 for error messages.

- clogHistMsgName—Mnemonic that identifies this SYSLOG message, up to 30 characters. If the mnemonic is longer than 30 characters, it is truncated to 29 characters and an asterisk (*) is appended to the end of the message to indicate that it has been truncated.
- clogHistMsgText—Actual text of the SYSLOG message, up to 255 characters, as it would appear in the console and SYSLOG logs. If a message is longer than 255 characters, it is truncated to 254 characters and an asterisk (*) is appended to the end of the message to indicate it has been truncated.
- clogHistTimestamp—Time stamp, in terms of sysUpTime, for when the SYSLOG message was generated.

Control Objects

The following objects in the CISCO-SYSLOG-MIB control the number and type of messages that are stored in the clogHistoryTable:

- clogMaxSeverity—Maximum severity level for the SYSLOG messages that are processed by this MIB. The default is 5 (warning), which indicates that SYSLOG messages of severity levels 1 through 5 are processed by the MIB.
- clogMsgIgnores—Number of SYSLOG messages that were ignored because it had a severity level greater than that specified by the clogMaxSeverity.
- clogMsgDrops—Number of SYSLOG messages that were dropped and not stored in the clogHistoryTable because of a lack of resources.
- clogHistTableMaxLength—Maximum number of SYSLOG messages that can be stored in the clogHistoryTable. When the table is full, the oldest message in the table is deleted to make room when a new SYSLOG message is generated. The valid range is 0 to 500, with a default of 1.
- clogHistMsgsFlushed—Number of entries that have been removed from the clogHistoryTable to make room for new entries. If this object is continually increasing, it indicates that you either need to increase the size of the table (clogHistTableMaxLength) or need to poll the table more frequently.

SYSLOG Notifications

You can configure the Cisco ASR 901 and 901S routers such that they generate an SNMP notification when a SYSLOG message is generated. The notification sends an clogMessageGenerated object, which contains the following objects that identify the SYSLOG message:

- clogHistFacility
- clogHistSeverity
- clogHistMsgName
- clogHistMsgText
- clogHistTimestamp

To enable SYSLOG notifications using CLI commands, give the following command in global configuration mode:

```
snmp-server enable traps syslog
```

To enable these notifications using SNMP commands, set the clogNotificationsEnabled object to true(1). The clogNotificationsSent object then contains the number of clogMessageGenerated notifications that have been sent.

Example

The following example shows typical output from the CISCO-SYSLOG-MIB when using the SNMP utilities that are standard on many UNIX-based systems. This router uses the default configuration, where only one SYSLOG message is stored in the clogHistoryTable. The table currently contains an entry with the index of 25, and clogHistMsgsFlushed shows that the 24 previous messages have already been flushed from the table.

```
csh% getmany -v2c 10.10.11.12 public ciscoSyslogMIB
```