Using Cisco RFGW-10 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 a Cisco RFGW-10.

Tips and Guidelines

When using SNMP to manage the Cisco RFGW-10, be aware of the following points.

IF-MIB Caching

In this release, the Cisco RFGW-10 implements 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. This improves performance, because it means the Cisco IOS or IOS-XE software does not have to poll each line card to obtain these counters when an SNMP request is made.

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 RFGW-10 using SNMP commands:

- 32-bit counters—A 32-bit counter wraps around to zero after reaching approximately 4.2 billion. On a busy RFGW-10, 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 RFGW-10 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 interface (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.

Route Processor Redundancy (RPR) Modules on the Cisco RFGW-10

On a Cisco RFGW-10 running Cisco IOS Release 12.2(44)SQ, SNMP configuration commands and CLI commands are not synchronized to the standby Supervisor. Hence, the active configuration is lost when a switchover occurs and the standby Supervisor is active. When the RFGW-10 switches back to the original Supervisor, the original configuration is restored.

Use CLI commands for critical configurations and save the configuration to the startup-config to ensure that the critical configuration is active during any switchovers.

Obtaining Basic Information About the RFGW-10

Basic information about the Cisco RFGW-10 can be obtained from objects in the following MIBs:

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

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 RFGW-10, 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 = STRING: ''NWG121902R6''

sysDescr.0 = STRING: Cisco IOS Software, Catalyst 4500 L3 Switch Software
(RFGW-10-ENTSERVICESK9-M), Version 12.2(122_44_SQ_20081124)SQ EARLY DEPLOYMENT
DATECODE BUILD, synced to V122_43_90_SQ
Copyright (c) 1986-2009 by Cisco Systems, Inc.
Compiled Mon 24-Nov-08 13:58 by jdker

sysObjectID.0 = ciscoProducts.ciscoCatRfgw
sysName.0 = RFGW-10
```

SNMPv2-MIB

The following objects in the SNMPv2-MIB provide basic information about the RFGW-10, its software, and other run-time information:

- sysDescr—Provides an overall description of the RFGW-10, including its model number and the version of Cisco IOS or IOS-XE software that it is running. For example:

```
csh% getmany -v2c 10.10.11.12 public sysDescr

sysDescr.0 = Cisco Internetwork Operating System Software
IOS (tm) 10000 Software (RFGW-10-K8P6-M), Released Version 12.2(15)BC1
Copyright (c) 1986-2009 by cisco Systems, Inc.
Compiled Fri 23-Jan-09 23:56 by atifg
```

- 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.ciscoRFGW-10
```

- sysName—Provides the host name for the RFGW-10, as assigned by the **hostname** command. For example:

```
csh% getmany -v2c 10.10.11.12 public sysName

sysName.0 = RFGW-10
```

- sysUpTime—Provides the time, in hundredths of a second, since the RFGW-10 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 RFGW-10, 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 RFGW-10'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 RFGW-10 hardware:

- entPhysicalDescr—Provides a description of each hardware component in the RFGW-10. For example, the following is a typical description for the Cisco RFGW-10 chassis:

```
csh% getnext -v2c 10.10.11.12 public entPhysicalDescr

entPhysicalDescr.1 = Cisco Systems, Inc. RFGW-10 14 slot switch
```

- 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 = NWG121902R6
```

- 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 = RFGW-10
```

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

Managing Physical Components

The Cisco RFGW-10 supports a number of MIBs for the management of the 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
- Monitors and configures the status of field-replaceable units (FRUs)
- 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 RFGW-10:

- ENTITY-MIB, page A-5
- Performing Inventory Management, page A-6

Tip To retrieve the chassis serial number for the RFGW-10, 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.

ENTITY-MIB

The Cisco RFGW-10 uses 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 card, a port on a card, a major subsystem on a card, a slot in the chassis, a field-replaceable unit (FRU), or any other equipment that is installed in the RFGW-10.

The ENTITY-MIB defines a set of objects that uniquely identify each entity in the RFGW-10, 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 RFGW-10:

- entPhysicalTable—Describes each physical component (entity) in the RFGW-10. 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 RFGW-10, which shows the relationship between physical entities in the chassis.

  Typically, the container tree is organized as follows:

  - The chassis is the topmost level and contains the processor card and chassis slots.
  - Chassis slots contain the individual line cards and I/O controller (if installed).
  - Line cards contain ports (interfaces).
  - RFGW-10 interface line cards contain downstream ports.

Performing Inventory Management

The ENTITY-MIB provides all of the information needed to collect an inventory of the physical components in the RFGW-10. The following procedure illustrates one way this can be done, using a RFGW-10. In this example, the RFGW-10 contains the following cards:

Slot 1: RFGW-10 Supervisor V-10GE, 2x10GE(X2) and 4x1GE(SFP)

Slot 2: RFGW-10 Supervisor V-10GE, 2x10GE(X2) and 4x1GE(SFP)

Slot 3: RFGW-10 Universal Downstram EQAM Card, 12 RF ports, 48 QAMs

Slot 13: RFGW-10 Timing, Communication, and Control Card

Slot 14: RFGW-10 Timing, Communication, and Control Card

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. For example:

```
entPhysicalDescr.1 = Cisco Systems, Inc. Cable-RFGW-10 14 slot switch
entPhysicalDescr.2 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.3 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.4 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.5 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.6 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.7 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.8 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.9 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.10 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.11 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.12 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.13 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.14 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.15 = Cable-RFGW-10 14 slot switch chassis slot
entPhysicalDescr.16 = Cable-RFGW-10 14 slot switch backplane
entPhysicalDescr.17 = Chassis Temperature Sensor
entPhysicalDescr.18 = Container of Fan Tray
entPhysicalDescr.19 = FanTray
entPhysicalDescr.20 = Container of Container of Power Supply
entPhysicalDescr.21 = Container of Power Supply
entPhysicalDescr.22 = Power Supply ( DC 4500W )
entPhysicalDescr.23 = Power Supply Fan Sensor
entPhysicalDescr.23 = Power Supply Fan Sensor
entPhysicalDescr.24 = Container of Power Supply
entPhysicalDescr.1000 = Supervisor V-10GE with 2 10GE X2 ports, and 4 1000BaseX SFP ports
entPhysicalDescr.1001 = Port Container
entPhysicalDescr.1002 = Port Container
entPhysicalDescr.1003 = Port Container
entPhysicalDescr.1004 = Port Container
entPhysicalDescr.1005 = Port Container
entPhysicalDescr.1006 = Port Container
entPhysicalDescr.1007 = 10Gbase-SR
entPhysicalDescr.1009 = 1000BaseSX
entPhysicalDescr.1010 = 1000BaseSX
entPhysicalDescr.2000 = Supervisor V-10GE with 2 10GE X2 ports, and 4 1000BaseX SFP ports
entPhysicalDescr.2001 = Port Container
entPhysicalDescr.2002 = Port Container
entPhysicalDescr.2003 = Port Container
entPhysicalDescr.2004 = Port Container
entPhysicalDescr.2005 = Port Container
entPhysicalDescr.2006 = Port Container
entPhysicalDescr.2007 = 10Gbase-LR
```

```
entPhysicalDescr.2009 = 1000BaseT
entPhysicalDescr.2010 = 1000BaseT
entPhysicalDescr.3000 = 48 QAM with 2 SFP(1000BaseX), 1 ASI
entPhysicalDescr.3001 = Port Container
entPhysicalDescr.3002 = Port Container
entPhysicalDescr.3005 = 1000BaseSX
entPhysicalDescr.3006 = 1000BaseSX
entPhysicalDescr.3007 = QAM port
entPhysicalDescr.3008 = QAM port
entPhysicalDescr.3009 = QAM port
entPhysicalDescr.3010 = QAM port
entPhysicalDescr.3011 = QAM port
entPhysicalDescr.3012 = QAM port
entPhysicalDescr.3013 = QAM port
entPhysicalDescr.3014 = QAM port
entPhysicalDescr.3015 = QAM port
entPhysicalDescr.3016 = QAM port
entPhysicalDescr.3017 = QAM port
entPhysicalDescr.3018 = QAM port
entPhysicalDescr.3019 = QAM channel
entPhysicalDescr.3020 = QAM channel
entPhysicalDescr.3021 = QAM channel
entPhysicalDescr.3022 = QAM channel
entPhysicalDescr.3023 = QAM channel
entPhysicalDescr.3024 = QAM channel
entPhysicalDescr.3025 = QAM channel
entPhysicalDescr.3026 = QAM channel
entPhysicalDescr.3027 = QAM channel
entPhysicalDescr.3028 = QAM channel
entPhysicalDescr.3029 = QAM channel
entPhysicalDescr.3030 = QAM channel
entPhysicalDescr.3031 = QAM channel
entPhysicalDescr.3032 = QAM channel
entPhysicalDescr.3033 = QAM channel
entPhysicalDescr.3034 = QAM channel
entPhysicalDescr.3035 = QAM channel
entPhysicalDescr.3036 = QAM channel
entPhysicalDescr.3037 = QAM channel
entPhysicalDescr.3038 = QAM channel
entPhysicalDescr.3039 = QAM channel
entPhysicalDescr.3040 = QAM channel
entPhysicalDescr.3041 = QAM channel
entPhysicalDescr.3042 = QAM channel
entPhysicalDescr.3043 = QAM channel
entPhysicalDescr.3044 = QAM channel
entPhysicalDescr.3045 = QAM channel
entPhysicalDescr.3046 = QAM channel
entPhysicalDescr.3047 = QAM channel
entPhysicalDescr.3048 = QAM channel
entPhysicalDescr.3049 = QAM channel
entPhysicalDescr.3050 = QAM channel
entPhysicalDescr.3051 = QAM channel
entPhysicalDescr.3052 = QAM channel
entPhysicalDescr.3053 = QAM channel
entPhysicalDescr.3054 = QAM channel
entPhysicalDescr.3055 = QAM channel
entPhysicalDescr.3056 = QAM channel
entPhysicalDescr.3057 = QAM channel
entPhysicalDescr.3058 = QAM channel
entPhysicalDescr.3059 = QAM channel
entPhysicalDescr.3060 = QAM channel
entPhysicalDescr.3061 = QAM channel
entPhysicalDescr.3062 = QAM channel
entPhysicalDescr.3063 = QAM channel
```

```
entPhysicalDescr.3064 = QAM channel
entPhysicalDescr.3065 = QAM channel
entPhysicalDescr.3066 = QAM channel
entPhysicalDescr.13000 = Timing Clock card (2 DTI ports)
entPhysicalDescr.13002 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13003 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13004 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13005 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13006 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13007 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13008 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13009 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13010 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13011 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13012 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13013 = Rf Switch Card (12 rf ports)
entPhysicalDescr.13016 = TCC-DTI TI/E1 input port
entPhysicalDescr.13017 = TCC-DTI TI/E1 input port
entPhysicalDescr.14000 = Timing Clock card (2 DTI ports)
entPhysicalDescr.14016 = TCC-DTI TI/E1 input port
entPhysicalDescr.14017 = TCC-DTI TI/E1 input port
```

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. Table A-1 shows typical descriptions for the objects used in this example.

Table A-1 Sample entPhysicalDescr Objects and Descriptions

Index #	entPhysicalDescr[1]	entPhysicalVendorType	entPhysicalName	entPhysicalClass
1	Cisco Systems, Inc. Cable-RFGW-10 14 slot switch	cevChassisCatRfgw	Switch System	chassis(3)
2	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 1	container(5)
3	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 2	container(5)
4	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 3	container(5)
5	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 4	container(5)
6	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 5	container(5)
7	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 6	container(5)
8	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 7	container(5)
9	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 8	container(5)
10	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 9	container(5)
11	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 10	container(5)
12	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 11	container(5)
13	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 12	container(5)
14	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 13	container(5)
15	Cable-RFGW-10 14 slot switch chassis slot	cevContainerSlot	Slot 14	container(5)
16	Cable-RFGW-10 14 slot switch backplane	cevBackplaneCatRfgw	Backplane	backplane(4)
17	Chassis Temperature Sensor	cevSensorCat4kTemp	Chassis Temperature Sensor	sensor(8)

Table A-1 Sample entPhysicalDescr Objects and Descriptions (continued)

Index #	entPhysicalDescr[1]	entPhysicalVendorType	entPhysicalName	entPhysicalClass
18	Container of Fan Tray	cevContainerCat4kFanTrayBay	Fan Tray Bay	container(5)
19	FanTray	cevFanCat4kFanTray	Fan	fan(7)
20	Container of Container of Power Supply	cevContainerCat4kContainerPSBay	Container of Power Supply Bay	container(5)
21	Container of Power Supply	cevContainerCat4kPowerSupplyBay	Power Supply Bay 1	container(5)
22	Power Supply (DC 4500W)	cevPowerSupplyDc4500W	Power Supply 1	powerSupply(6)
23	Power Supply Fan Sensor	cevFanCat4kPowerSupplyFan	Power Supply 1 Fan	sensor(8)
24	Container of Power Supply	cevContainerCat4kPowerSupplyBay	Power Supply Bay 2	container(5)
1000	Supervisor V-10GE with 2 10GE X2 ports, and 4 1000BaseX SFP ports	cevCat4kWsx451610GE	Linecard(slot 1)	module(9)
1001	Port Container	cevContainer10GigBasePort	Port Container 1/1	container(5)
1002	Port Container	cevContainer10GigBasePort	Port Container 1/2	container(5)
1003	Port Container	cevContainer10GigBasePort	Port Container 1/3	container(5)
1004	Port Container	cevContainer10GigBasePort	Port Container 1/4	container(5)
1005	Port Container	cevContainer10GigBasePort	Port Container 1/5	container(5)
1006	Port Container	cevContainer10GigBasePort	Port Container 1/6	container(5)
1007	10Gbase-SR	cevMX210GBaseSR	TenGigabitEthernet1/1	port(10)
1009	1000BaseSX	cevMGBIC1000BaseSX	GigabitEthernet1/3	port(10)
1010	1000BaseSX	cevMGBIC1000BaseSX	GigabitEthernet1/4	port(10)
2000	Supervisor V-10GE with 2 10GE X2 ports, and 4 1000BaseX SFP ports	cevCat4kWsx451610GE	Linecard(slot 2)	module(9)
2001	Port Container	cevContainer10GigBasePort	Port Container 2/1	container(5)
2002	Port Container	cevContainer10GigBasePort	Port Container 2/2	container(5)
2003	Port Container	cevContainer10GigBasePort	Port Container 2/3	container(5)

Table A-1 Sample entPhysicalDescr Objects and Descriptions (continued)

Index #	entPhysicalDescr[1]	entPhysicalVendorType	entPhysicalName	entPhysicalClass
2004	Port Container	cevContainer10GigBasePort	Port Container 2/4	container(5)
2005	Port Container	cevContainer10GigBasePort	Port Container 2/5	container(5)
2006	Port Container	cevContainer10GigBasePort	Port Container 2/6	container(5)
2007	10Gbase-LR	cevMGBIC1000BaseSX	TenGigabitEthernet2/1	port(10)
2009	1000BaseT	cevMGBIC1000BaseT	GigabitEthernet2/3	port(10)
2010	1000BaseT	cevMGBIC1000BaseT	GigabitEthernet2/4	port(10)
3000	48 QAM with 2 SFP(1000BaseX), 1 ASI	cevRfgwQam48ABC	Linecard(slot 3)	module(9)
	Port Container	cevContainerSFP	Port Container 3/13	container(5)
3002	Port Container	cevContainerSFP	Port Container 3/14	container(5)
3005	1000BaseSX	cevMGBIC1000BaseSX	GigabitEthernet3/13	port(10)
3006	1000BaseSX	cevMGBIC1000BaseSX	GigabitEthernet3/14	port(10)
3007	QAM port	cevRfgwPort4Qam	Qam3/1	module(9)
3008	QAM port	cevRfgwPort4Qam	Qam3/2	module(9)
3009	QAM port	cevRfgwPort4Qam	Qam3/3	module(9)
3010	QAM port	cevRfgwPort4Qam	Qam3/4	module(9)
3011	QAM port	cevRfgwPort4Qam	Qam3/5	module(9)
3012	QAM port	cevRfgwPort4Qam	Qam3/6	module(9)
3013	QAM port	cevRfgwPort4Qam	Qam3/7	module(9)
3014	QAM port	cevRfgwPort4Qam	Qam3/8	module(9)
3015	QAM port	cevRfgwPort4Qam	Qam3/9	module(9)
3016	QAM port	cevRfgwPort4Qam	Qam3/10	module(9)
3017	QAM port	cevRfgwPort4Qam	Qam3/11	module(9)
3018	QAM port	cevRfgwPort4Qam	Qam3/12	module(9)
3019	QAM channel	cevPortRfDs	Qam3/1.1	port(10)
3020	QAM channel	cevPortRfDs	Qam3/1.2	port(10)
3021	QAM channel	cevPortRfDs	Qam3/1.3	port(10)
3022	QAM channel	cevPortRfDs	Qam3/1.4	port(10)
3023	QAM channel	cevPortRfDs	Qam3/2.1	port(10)
3024	QAM channel	cevPortRfDs	Qam3/2.2	port(10)
3025	QAM channel	cevPortRfDs	Qam3/2.3	port(10)
3026	QAM channel	cevPortRfDs	Qam3/2.4	port(10)

Table A-1 Sample entPhysicalDescr Objects and Descriptions (continued)

Index #	entPhysicalDescr[1]	entPhysicalVendorType	entPhysicalName	entPhysicalClass
3027	QAM channel	cevPortRfDs	Qam3/3.1	port(10)
3028	QAM channel	cevPortRfDs	Qam3/3.2	port(10)
3029	QAM channel	cevPortRfDs	Qam3/3.3	port(10)
3030	QAM channel	cevPortRfDs	Qam3/3.4	port(10)
3031	QAM channel	cevPortRfDs	Qam3/4.1	port(10)
3032	QAM channel	cevPortRfDs	Qam3/4.2	port(10)
3033	QAM channel	cevPortRfDs	Qam3/4.3	port(10)
3034	QAM channel	cevPortRfDs	Qam3/4.4	port(10)
3035	QAM channel	cevPortRfDs	Qam3/5.1	port(10)
3036	QAM channel	cevPortRfDs	Qam3/5.2	port(10)
3037	QAM channel	cevPortRfDs	Qam3/5.3	port(10)
3038	QAM channel	cevPortRfDs	Qam3/5.4	port(10)
3039	QAM channel	cevPortRfDs	Qam3/6.1	port(10)
3040	QAM channel	cevPortRfDs	Qam3/6.2	port(10)
3041	QAM channel	cevPortRfDs	Qam3/6.3	port(10)
3042	QAM channel	cevPortRfDs	Qam3/6.4	port(10)
3043	QAM channel	cevPortRfDs	Qam3/7.1	port(10)
3044	QAM channel	cevPortRfDs	Qam3/7.2	port(10)
3045	QAM channel	cevPortRfDs	Qam3/7.3	port(10)
3046	QAM channel	cevPortRfDs	Qam3/7.4	port(10)
3047	QAM channel	cevPortRfDs	Qam3/8.1	port(10)
3048	QAM channel	cevPortRfDs	Qam3/8.2	port(10)
3049	QAM channel	cevPortRfDs	Qam3/8.3	port(10)
3050	QAM channel	cevPortRfDs	Qam3/8.4	port(10)
3051	QAM channel	cevPortRfDs	Qam3/9.1	port(10)
3052	QAM channel	cevPortRfDs	Qam3/9.2	port(10)
3053	QAM channel	cevPortRfDs	Qam3/9.3	port(10)
3054	QAM channel	cevPortRfDs	Qam3/9.4	port(10)
3055	QAM channel	cevPortRfDs	Qam3/10.1	port(10)
3056	QAM channel	cevPortRfDs	Qam3/10.2	port(10)
3057	QAM channel	cevPortRfDs	Qam3/10.3	port(10)
3058	QAM channel	cevPortRfDs	Qam3/10.4	port(10)
3059	QAM channel	cevPortRfDs	Qam3/11.1	port(10)
3060	QAM channel	cevPortRfDs	Qam3/11.2	port(10)
3061	QAM channel	cevPortRfDs	Qam3/11.3	port(10)

Table A-1 Sample entPhysicalDescr Objects and Descriptions (continued)

Index #	entPhysicalDescr[1]	entPhysicalVendorType	entPhysicalName	entPhysicalClass
3062	QAM channel	cevPortRfDs	Qam3/11.4	port(10)
3063	QAM channel	cevPortRfDs	Qam3/12.1	port(10)
3064	QAM channel	cevPortRfDs	Qam3/12.2	port(10)
3065	QAM channel	cevPortRfDs	Qam3/12.3	port(10)
3066	QAM channel	cevPortRfDs	Qam3/12.4	port(10)
13000	Timing Clock card (2 DTI ports)	cevRfgwDtcc	Linecard(slot 13)	module(9)
13002	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)
13003	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)
13004	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)
13005	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)
13006	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)
13007	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)
13008	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)
13009	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)
13010	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)
13011	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)
13012	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)
13013	Rf Switch Card (12 rf ports)	cevRfgwRfswitch	Rf Switch Card (12 rf ports) card in linecard 13	other(1)

Table A-1 Sample entPhysicalDescr Objects and Descriptions (continued)

Index #	entPhysicalDescr[1]	entPhysicalVendorType	entPhysicalName	entPhysicalClass
13016	TCC-DTI TI/E1 input port	cevPortDti	Dti13/1	port(10)
13017	TCC-DTI TI/E1 input port	cevPortDti	Dti13/2	port(10)
14000	Timing Clock card (2 DTI ports)	cevRfgwDtcc	Linecard(slot 14)	module(9)
14016	TCC-DTI TI/E1 input port	cevPortDti	Dti14/1	port(10)
14017	TCC-DTI TI/E1 input port	cevPortDti	Dti14/2	port(10)

1. Interfaces are typically identified by the chipset that is being used for the interface's connectors. On cable interfaces, upstreams are further identified by the chipsets that are providing the PHY-layer connectivity.

Step 3 To create the containment tree for the RFGW-10, 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.

Table A-2 shows the parent container for the entPhysicalDescr objects being used in this example:

Table A-2 Relationship of EntPhysicalContainedIn to entPhysicalDescr

#	entPhysicalDescr	entPhysicalContainedInValue	Parent Container
1	Cisco Systems, Inc. Cable-RFGW-10 14 slot switch	0	
2	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
3	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
4	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
5	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
6	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
7	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
8	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
9	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
10	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
11	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
12	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
13	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
14	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
15	Cable-RFGW-10 14 slot switch chassis slot	1	Chassis
16	Cable-RFGW-10 14 slot switch backplane	1	Chassis
17	Chassis Temperature Sensor	1	Chassis
18	Container of Fan Tray		Chassis
19	FanTray	18	Container
20	Container of Container of Power Supply	1	Chassis

Table A-2 Relationship of EntPhysicalContainedIn to entPhysicalDescr (continued)

#	entPhysicalDescr	entPhysicalContainedInValue	Parent Container
21	Container of Power Supply	20	Container
22	Power Supply (DC 4500W)	21	Container of Power Supply
23	Power Supply Fan Sensor	22	Power Supply
24	Container of Power Supply	20	Container of Container of Power Supply
1000	Supervisor V-10GE with 2 10GE X2 ports, and 4 1000BaseX SFP ports	2	Chassis slot
1001	Port Container	1000	Supervisor
1002	Port Container	1000	Supervisor
1003	Port Container	1000	Supervisor
1004	Port Container	1000	Supervisor
1005	Port Container	1000	Supervisor
1006	Port Container	1000	Supervisor
1007	10Gbase-SR	1001	Port Container
1009	1000BaseSX	1003	Port Container
1010	1000BaseSX	1004	Port Container
2000	Supervisor V-10GE with 2 10GE X2 ports, and 4 1000BaseX SFP ports	3	Chassis slot
2001	Port Container	2000	Supervisor
2002	Port Container	2000	Supervisor
2003	Port Container	2000	Supervisor
2004	Port Container	2000	Supervisor
2005	Port Container	2000	Supervisor
2006	Port Container	2000	Supervisor
2007	10Gbase-LR	2001	Port Container
2009	1000BaseT	2003	Port Container
2010	1000BaseT	2004	Port Container
3000	48 QAM with 2 SFP(1000BaseX), 1 ASI	4	Chassis slot
3001	Port Container	3000	DS-48 Line Card
3002	Port Container	3000	DS-48 Line Card
3005	1000BaseSX	3001	Port Container
3006	1000BaseSX	3002	Port Container

Table A-2 Relationship of EntPhysicalContainedIn to entPhysicalDescr (continued)

#	entPhysicalDescr	entPhysicalContainedInValue	Parent Container
3007	QAM port	3000	DS-48 Line Card
3008	QAM port	3000	DS-48 Line Card
3009	QAM port	3000	DS-48 Line Card
3010	QAM port	3000	DS-49 Line Card
3011	QAM port	3000	DS-50 Line Card
3012	QAM port	3000	DS-51 Line Card
3013	QAM port	3000	DS-52 Line Card
3014	QAM port	3000	DS-53 Line Card
3015	QAM port	3000	DS-54 Line Card
3016	QAM port	3000	DS-55 Line Card
3017	QAM port	3000	DS-56 Line Card
3018	QAM port	3000	DS-57 Line Card
3019	QAM channel	3007	QAM port
3020	QAM channel	3007	QAM port
3021	QAM channel	3007	QAM port
3022	QAM channel	3007	QAM port
3023	QAM channel	3008	QAM port
3024	QAM channel	3008	QAM port
3025	QAM channel	3008	QAM port
3026	QAM channel	3008	QAM port
3027	QAM channel	3009	QAM port
3028	QAM channel	3009	QAM port
3029	QAM channel	3009	QAM port
3030	QAM channel	3009	QAM port
3031	QAM channel	3010	QAM port
3032	QAM channel	3010	QAM port

Table A-2 Relationship of EntPhysicalContainedIn to entPhysicalDescr (continued)

#	entPhysicalDescr	entPhysicalContainedInValue	Parent Container
3033	QAM channel	3010	QAM port
3034	QAM channel	3010	QAM port
3035	QAM channel	3011	QAM port
3036	QAM channel	3011	QAM port
3037	QAM channel	3011	QAM port
3038	QAM channel	3011	QAM port
3039	QAM channel	3012	QAM port
3040	QAM channel	3012	QAM port
3041	QAM channel	3012	QAM port
3042	QAM channel	3012	QAM port
3043	QAM channel	3013	QAM port
3044	QAM channel	3013	QAM port
3045	QAM channel	3013	QAM port
3046	QAM channel	3013	QAM port
3047	QAM channel	3014	QAM port
3048	QAM channel	3014	QAM port
3049	QAM channel	3014	QAM port
3050	QAM channel	3014	QAM port
3051	QAM channel	3015	QAM port
3052	QAM channel	3015	QAM port
3053	QAM channel	3015	QAM port
3054	QAM channel	3015	QAM port
3055	QAM channel	3016	QAM port
3056	QAM channel	3016	QAM port
3057	QAM channel	3016	QAM port
3058	QAM channel	3016	QAM port
3059	QAM channel	3017	QAM port
3060	QAM channel	3017	QAM port
3061	QAM channel	3017	QAM port
3062	QAM channel	3017	QAM port
3063	QAM channel	3018	QAM port
3064	QAM channel	3018	QAM port
3065	QAM channel	3018	QAM port
3066	QAM channel	3018	QAM port
13000	Timing Clock card (2 DTI ports)	14	Chassis slot

Table A-2 Relationship of EntPhysicalContainedIn to entPhysicalDescr (continued)

#	entPhysicalDescr	entPhysicalContainedInValue	Parent Container
13002	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13003	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13004	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13005	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13006	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13007	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13008	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13009	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13010	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13011	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13012	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13013	Rf Switch Card (12 rf ports)	13000	Timing Clock card
13016	TCC-DTI TI/E1 input port	13000	Timing Clock card
13017	TCC-DTI TI/E1 input port	13000	Timing Clock card
14000	Timing Clock card (2 DTI ports)	15	Chassis slot
14016	TCC-DTI TI/E1 input port	14000	Timing Clock card
14017	TCC-DTI TI/E1 input port	14000	Timing Clock card

Step 4 (Optional)If a parent object contains multiple children that are the same type of object, such as a RFGW-10 that contains multiple line card slots (Chassis Slots), 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.

Table A-3 shows how the entPhysicalDescr objects that refer to chassis slots can be put into their physical order by using their entPhysicalParentRelPos values. For example, entPhysicalDescr.4 has an entPhysicalParentRelPos value of 3, which indicates that this slot is slot 3 in the RFGW-10 chassis.

Table A-3 Using entPhysicalParentRelPos to Order entPhysicalDescr Objects

#	entPhysicalDescr	entPhysicalContainedIn	entPhysicalParentRelPos	Physical Slot #
1	Cisco Systems, Inc. Cable-RFGW-10 14 slot switch	0 = TopLevel	-1	N/A
2	Cable-RFGW-10 14 slot switch chassis slot	1 = Chassis	1	Slot 1
3	Cable-RFGW-10 14 slot switch chassis slot	2 = Chassis	2	Slot 2
4	Cable-RFGW-10 14 slot switch chassis slot	3 = Chassis	3	Slot 3
5	Cable-RFGW-10 14 slot switch chassis slot	4 = Chassis	4	Slot 4
6	Cable-RFGW-10 14 slot switch chassis slot	5 = Chassis	5	Slot 5
7	Cable-RFGW-10 14 slot switch chassis slot	6 = Chassis	6	Slot 6
8	Cable-RFGW-10 14 slot switch chassis slot	7 = Chassis	7	Slot 7
9	Cable-RFGW-10 14 slot switch chassis slot	8 = Chassis	8	Slot 8
10	Cable-RFGW-10 14 slot switch chassis slot	9 = Chassis	9	Slot 9
11	Cable-RFGW-10 14 slot switch chassis slot	10 = Chassis	10	Slot 10
12	Cable-RFGW-10 14 slot switch chassis slot	11 = Chassis	11	Slot 11

Table A-3 Using entPhysicalParentRelPos to Order entPhysicalDescr Objects (continued)

#	entPhysicalDescr	entPhysicalContainedIn	entPhysicalParentRelPos	Physical Slot #
13	Cable-RFGW-10 14 slot switch chassis slot	12 = Chassis	12	Slot 12
14	Cable-RFGW-10 14 slot switch chassis slot	13 = Chassis	13	Slot 13
15	Cable-RFGW-10 14 slot switch chassis slot	14 = Chassis	14	Slot 14

Step 5 (Optional)To map a physical interface to its ifIndex, which is defined in IF-MIB and used in other MIBs to uniquely identify a logical interface, use the entAliasMappingIdentifier object.

For example, the following shows the entAliasMappingIdentifier values for the RFGW-10 used in this example. In this example, entPhysicalDescr.3019 (which Table A-1 identifies as the Qam3/1.1 interface) maps to an ifIndex value of 199.

```
entAliasMappingIdentifier.1007.0 = ifIndex.3
entAliasMappingIdentifier.1009.0 = ifIndex.5
entAliasMappingIdentifier.1010.0 = ifIndex.6
entAliasMappingIdentifier.2007.0 = ifIndex.9
entAliasMappingIdentifier.2009.0 = ifIndex.11
entAliasMappingIdentifier.2010.0 = ifIndex.12
entAliasMappingIdentifier.3005.0 = ifIndex.15
entAliasMappingIdentifier.3006.0 = ifIndex.16
entAliasMappingIdentifier.3019.0 = ifIndex.199
entAliasMappingIdentifier.3020.0 = ifIndex.200
entAliasMappingIdentifier.3021.0 = ifIndex.201
entAliasMappingIdentifier.3022.0 = ifIndex.202
entAliasMappingIdentifier.3023.0 = ifIndex.203
entAliasMappingIdentifier.3024.0 = ifIndex.204
entAliasMappingIdentifier.3025.0 = ifIndex.205
entAliasMappingIdentifier.3026.0 = ifIndex.206
entAliasMappingIdentifier.3027.0 = ifIndex.207
entAliasMappingIdentifier.3028.0 = ifIndex.208
entAliasMappingIdentifier.3029.0 = ifIndex.209
entAliasMappingIdentifier.3030.0 = ifIndex.210
entAliasMappingIdentifier.3031.0 = ifIndex.211
entAliasMappingIdentifier.3032.0 = ifIndex.212
entAliasMappingIdentifier.3033.0 = ifIndex.213
entAliasMappingIdentifier.3034.0 = ifIndex.214
entAliasMappingIdentifier.3035.0 = ifIndex.215
entAliasMappingIdentifier.3036.0 = ifIndex.216
entAliasMappingIdentifier.3037.0 = ifIndex.217
entAliasMappingIdentifier.3038.0 = ifIndex.218
entAliasMappingIdentifier.3039.0 = ifIndex.219
entAliasMappingIdentifier.3040.0 = ifIndex.220
entAliasMappingIdentifier.3041.0 = ifIndex.221
entAliasMappingIdentifier.3042.0 = ifIndex.222
entAliasMappingIdentifier.3043.0 = ifIndex.223
entAliasMappingIdentifier.3044.0 = ifIndex.224
entAliasMappingIdentifier.3045.0 = ifIndex.225
entAliasMappingIdentifier.3046.0 = ifIndex.226
entAliasMappingIdentifier.3047.0 = ifIndex.227
entAliasMappingIdentifier.3048.0 = ifIndex.228
entAliasMappingIdentifier.3049.0 = ifIndex.229
```

```
entAliasMappingIdentifier.3050.0 = ifIndex.230
entAliasMappingIdentifier.3051.0 = ifIndex.231
entAliasMappingIdentifier.3052.0 = ifIndex.232
entAliasMappingIdentifier.3053.0 = ifIndex.233
entAliasMappingIdentifier.3054.0 = ifIndex.234
entAliasMappingIdentifier.3055.0 = ifIndex.235
entAliasMappingIdentifier.3056.0 = ifIndex.236
entAliasMappingIdentifier.3057.0 = ifIndex.237
entAliasMappingIdentifier.3058.0 = ifIndex.238
entAliasMappingIdentifier.3059.0 = ifIndex.239
entAliasMappingIdentifier.3060.0 = ifIndex.240
entAliasMappingIdentifier.3061.0 = ifIndex.241
entAliasMappingIdentifier.3062.0 = ifIndex.242
entAliasMappingIdentifier.3063.0 = ifIndex.243
entAliasMappingIdentifier.3064.0 = ifIndex.244
entAliasMappingIdentifier.3065.0 = ifIndex.245
entAliasMappingIdentifier.3066.0 = ifIndex.246
entAliasMappingIdentifier.13016.0 = ifIndex.189
entAliasMappingIdentifier.13017.0 = ifIndex.190
entAliasMappingIdentifier.14016.0 = ifIndex.191
entAliasMappingIdentifier.14017.0 = ifIndex.192
```

Generating SNMP Traps

This section describes how to configure the Cisco RFGW-10 to generate SNMP traps when certain events or conditions occur on the RFGW-10. To use SNMP commands to configure the RFGW-10 to generate SNMP traps, you must define at least one target host to receive the traps, using the following procedure:

Tip You can also use the command-line interface (CLI) to enable and configure the generation of traps on the RFGW-10. For information on using the CLI, see the "Enabling Notifications" section on page 4-2.

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 RFGW-10 waits for a response from the host (if any). The default is 1500 (15 seconds).
- snmpTargetAddrRetryCount—Default number of times that the Cisco RFGW-10 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 RFGW-10 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 RFGW-10 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.

Step 3 Create an entry in the snmpNotifyTable, which is defined in the SNMP-NOTIFICATION-MIB. Each row in this table contains the following objects, which define a set of host targets that are to receive traps:

- snmpNotifyName—Unique string, up to 32 characters, that identifies this particular row entry.
- snmpNotifyTag—Arbitrary string, up to 255 characters, that identifies the set of hosts to receive traps. This tag value is matched against the snmpTargetAddrTagList object to determine which hosts should receive which traps.
- snmpNotifyType—Defines the type of trap to be set: trap(1) or inform(2). The default is trap(1).
- snmpNotifyStorageType—Type of storage to be used for this row entry: volatile(2), nonVolatile(3), permanent(4), or readOnly(5). The default is nonVolatile(4).
- snmpNotifyRowStatus—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 4 Optionally create rows in the snmpNotifyFilterProfileTable and snmpNotifyFilterTable, which are defined in the SNMP-NOTIFICATION-MIB. These tables create notification filters that limit the types of notifications that the RFGW-10 sends to particular hosts.

Step 5 Optionally enable traps and notifications to be sent. Most other MIBs include their own objects of NOTIFICATION-TYPE that enable or disable feature-specific traps. These notification objects also define the varbinds that are sent with each trap, which contain the specific information about the event that occurred.

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

Table A-4 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 RFGW-10 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.
Alarm is Asserted or Cleared	`snmp-server enable traps alarms`	When ALARM traps are enabled, the RFGW-10 generates a trap whenever an alarm is asserted or cleared for physical entities that are defined in the entPhysicalTable in the ENTITY-MIB.

Identifying Cisco Unique Device Identifiers

In order to use UDI retrieval, the Cisco product in use must be UDI-enabled. A UDI-enabled Cisco product supports five required Entity MIB objects. The five Entity MIB v2 (RFC-2737) objects are as follows:

- entPhysicalName
- entPhysicalDescr
- entPhysicalModelName
- entPhysicalHardwareRev
- entPhysicalSerialNum

Although the show inventory command may be available, using that command on devices that are not UDI-enabled produces no output.

Before using the UDI Retrieval feature, you should understand the following concepts:

Unique Device Identifier Overview-Each identifiable product is an entity, as defined by the Entity MIB (RFC-2737) and its supporting documents. Some entities, such as a chassis, will have subentities like slots. An Ethernet switch might be a member of a superentity such as a stack. Most Cisco entities that are orderable products will leave the factory with an assigned UDI.

The UDI information is printed on a label that is affixed to the physical hardware device, and it is also stored electronically on the device in order to facilitate remote retrieval.

A UDI consists of the following elements:

- Product identifier (PID) The PID is the name by which the product can be ordered; it has been historically called the Product Name or Part Number. This is the identifier that one would use to order an exact replacement part.

- Version identifier (VID) The VID is the version of the product. Whenever a product has been revised, the VID will be incremented. The VID is incremented according to a rigorous process derived from Telcordia GR-209-CORE, an industry guideline that governs product change notices.
- Serial number (SN) The SN is the vendor-unique serialization of the product. Each manufactured product will carry a unique serial number assigned at the factory, which cannot be changed in the field. This is the means by which to identify an individual, specific instance of a product.