

Examples

Card Configuration

You can configure network and service cards by issuing SNMP set instructions to various objects in the VCO/4K MIB card tables. The following functions are possible:

- Add a card
- Delete a card
- Get the status of a card
- Modify card parameters



Note

All the commands in this chapter and throughout this document are from a hypothetical command set and not from a specific application.

Specifying the Index Value

Before using the MIB to perform card functions, you must know the index value of the card. Refer to the “Formula for Determining Index Value” section on page 1-6 for information on determining card index values.

Adding a Card

The following subsections give the general command syntax for adding single, dual, and triple index cards.

Single Index

Use the following steps to add a single index card:

-
- | | |
|--------|--|
| Step 1 | Set the card EntryStatus (specifying the location) to createRequest (2). |
| Step 2 | Set the card EntryStatus (specifying the location) to valid (1). |
-

Command Format

```
SNMP_SET <card>EntryStatus.<location> createRequest
SNMP_SET <card>EntryStatus.<location> valid
```

Example

To add a line card at slot 1-1-5, use the following syntax to enter the command:

```
SNMP_SET lcEntryStatus.5 integer 2
SNMP_SET lcEntryStatus.5 integer 1
```

Dual Index

Use the following steps to add a dual index card:

-
- | | |
|---------------|---|
| Step 1 | Set the card EntryStatus (specifying the location and type) to createRequest (2). |
| Step 2 | Set the card EntryStatus (specifying the location and type) to valid (1). |
-

Command Format

```
SNMP_SET <card>EntryStatus.<location>.<type> createRequest
SNMP_SET <card>EntryStatus.<location>.<type> valid
```

Example

To add a PRI/N card at slot 1-1-5, enter the following command syntax:

```
SNMP_SET isdnEntryStatus.5.2 integer 2
SNMP_SET isdnEntryStatus.5.2 integer 1
```

Triple Index

Triple index cards include any multispan network interface cards, including 4xT1, 4xE1, and ICC 16-span E1 or T1 I/O. The MVDC is also considered a triple index card.

Use the following steps to add a triple index card:

-
- | | |
|---------------|---|
| Step 1 | Set the EntryStatus (specifying the location, type, and span) to createRequest (2). |
| Step 2 | Set the EntryStatus (specifying the location, type, and span) to valid (1). |
-

Command Format

```
SNMP_SET <ptc_card>EntryStatus.<location>.<type>.<span> createRequest
```

```
SNMP_SET <ptc_card>EntryStatus.<location>.<type>.<span> valid
```

Example

```
SNMP_SET ptcEntryStatus.4.1.1 integer 2
```

```
SNMP_SET ptcEntryStatus.4.1.1 integer 1
```

Considerations

Consider the following when adding a card to the database:

- When you add a card, the initial status of the card is Out-of-Service (OOS). Use the CardStatus object to bring the card on line and make it available for the system to use.
- When you add an NBC, the card must use an R-L-S address of 1-1-2. Other hardware addresses are not valid for an NBC.
- Except for DTG, DRC 24, and DRC 48 cards, each port on a card requires one time-slot. When you add a new card to the database, the number of available time-slots is reduced by the number of ports on the card.
- When you add a new card and the number of ports on the card cause the number of ports to exceed your system's time-slot capacity, an error status is returned. You are unable to add this or any other card to the system until you remove existing cards or upgrade your Time-Slot Capacity License. (Contact your Cisco Systems sales representative for information on upgrading your license.)

Deleting a Card



Note

The command for deleting single, dual, or triple index cards is the same, except for the index value.

To delete a card, set the EntryStatus object to invalid (4).

Command Format

```
SNMP_SET <card>EntryStatus.<location> invalid
```

Example

```
SNMP_SET lcEntryStatus.2 integer 4
```

This command shown in this example deletes the line card at slot 1-1-2.

Getting the Card Status

To get the card status, enter an SNMP_GET command on the status of the card object.

Command Format

```
SNMP_GET <card>Status.<location>
```

Example

```
SNMP_GET lcStatus.2
```

The command shown in this example returns the status of the line card at slot 1-1-2. If the line card does not exist (for example, if you had deleted it, or the card was never configured), the value returned is an error. If the card exists, the state of the card is returned. Refer to Appendix A, “Card Error Messages” for a list of card error messages.

Modifying Card Parameters

To modify card parameters, enter an SNMP_GET command to see all the read-write parameters. Decide which parameters need modifying, and perform the following steps:

-
- Step 1 Set the EntryStatus object to underModification (3).
 - Step 2 Set the new parameter values.
 - Step 3 Set the EntryStatus object to valid (1).

To confirm that the parameters were changed, enter an SNMP_GET command.

Command Format

```
SNMP_SET <card>EntryStatus.<location> underModification
```

```
SNMP_SET <ObjectName>.<location>.<type> <value>
```

```
SNMP_SET <card>EntryStatus.<location> valid
```

Example

To activate a T1 card at slot 1-1-4 enter the following command syntax:

```
SNMP_SET tcEntryStatus.4.1 integer 3
```

```
SNMP_SET tcStatus.4.1 integer 1
```

```
SNMP_SET tcEntryStatus.4.1 integer 1
```

The commands in this example have the following actions:

- The first command declares that you are modifying something in the trunk card table. This is an indication to other cooperating management applications that you are modifying the table entry. You should also set the OwnerString, although it is not required.

- The second command declares that you are setting the status value in the trunk card to 1 (active).
- The third command declares that you are setting the EntryStatus value in the trunk card to valid (1). This setting verifies your changes and releases the trunk card object.

The following special considerations are associated with card status changes:

- Cards can be set to Active, Maintenance, Out-of-Service, Diagnostic, Remote Loopback, or Payload Loopback.
- A card must be in the database before you can make a status change.
- If you change a card to Active or Maintenance status, it must be physically installed in the system.
- You can change a card to Diagnostic status only if it is already in Active or Maintenance status.
- When you change a card to Out-of-Service status, any calls into ports that are linked on that card are torn down.
- To reset an active card, change the status from Active to Out-of-Service and then back to Active.
- When an IPRC is changed to Active mode, the voice announcement application is automatically downloaded to it from hard disk, if necessary. IPRCs are only downloaded following a power-up reset or if the application on the card does not match the download file on the system disk. If a download is not successful, the card remains in Maintenance mode. If download is successful, the voice prompt information is downloaded and the IPRC goes to Active.
- The IPRC table provides MIB objects where you can specify the length of time to chop the end of a newly recorded prompt, the interval timer for checksum verification, and the prompt libraries supported by the card. The IPRC Card Configuration Table also contains objects that record rack position and card operating state.

**Note**

Because you can reallocate time slots and remove prompt information loaded onto the IPRC, you must remove the IPRC from service prior to making any changes to the IPRC objects.

- The card configuration table also has objects that enable you to configure the port density in 8-port increments, thus reallocating time slots based on the defined port density. You can configure the 64- and 128-port IPRCs to support less than the physical port capacity.
- When you change a CPA to Active mode, the tone detection application is automatically downloaded to it from the hard disk, if necessary. CPAs are only downloaded following a power-up reset or if the application on the card does not match the download file on the system disk. If a download is not successful, the card remains in Maintenance (m) mode. If the download is successful, the CPA goes to Active mode.
- When you change the status to Diagnostic mode, the system checks whether any ports on the card are linked into a call. If one or more ports is active, the card is placed into Camped On mode (supported on T1, RC and TC cards). You can now run Diagnostic routines; the system skips any ports linked into a call.
- If you try to change from Camped On to Diagnostic mode, the system again checks if any ports on the card are still linked into a call.
- Remote Loopback mode applies only to PRI and Programmable Trunk cards. This mode enables you to run diagnostics on the total loop (trunks to the far end interface, and back to the VCO). The card must first be in Active mode before you can change its status to Remote Loopback.
- Payload Loopback applies only to Programmable T1 Trunk cards in ESF mode. In this mode the T1 stream is looped back to the network with a new frame.

Error Status

If any of the operations described in the preceding sections fails, enter an `SNMP_GET` command on the appropriate `ErrorStatus` object. Refer to the relevant card `ErrorStatus` objects in the index of this document.

Command Format

```
SNMP_GET <card>ErrorStatus
```

Example

```
SNMP_GET lcErrorStatus.4
```

In this example, the value returned tells you what the error is. These values are defined in the `ErrorStatus` object for each table in the MIB.

Port Configuration

You can configure network ports by issuing SNMP set instructions to various objects in the VCO/4K MIB card tables. The following functions are possible:

- Activate a port
- Deactivate a port
- Modify a port parameter
- Get the error status of a port

Activating a Port

To activate a port, perform the following steps:

-
- | | |
|---------------|---|
| Step 1 | Set the <code>PortEntryStatus</code> (specifying the location and port number) to <code>underModification</code> (3). |
| Step 2 | Set the <code>PortState</code> (specifying the location and port number) to <code>active</code> (1). |
| Step 3 | Set the <code>PortEntryStatus</code> (specifying the location and port number) to <code>valid</code> (1). |
-

Command Format

```
SNMP_SET <card>PortEntryStatus.<location>.<PortNumber> underModification
```

```
SNMP_SET <card>PortState.<location>.<PortNumber> active
```

```
SNMP_SET <card>PortEntryStatus.<location>.<PortNumber> valid
```

Example

To activate the first line card port at slot 1-1-3, use the following syntax to enter the command:

```
SNMP_SET lcPortEntryStatus.3.1 integer 3
```

```
SNMP_SET lcPortState.3.1 integer 1
```

```
SNMP_SET lcPortEntryStatus.3.1 integer 1
```

The commands in this example have the following actions:

- The first command sets the first line card port at slot 1-1-3 to underModification. This indicates to other cooperating management applications that you are modifying the port.
- The second command sets the state of the line card port to active.
- The third command sets the line card port to valid. This command verifies your settings and releases the port.

Deactivating a Port

To deactivate a port perform the following steps:

-
- | | |
|---------------|---|
| Step 1 | Set the PortEntryStatus (specifying the location and port number) to underModification (3). |
| Step 2 | Set the PortState (specifying the location and port number) to inactive (2). |
| Step 3 | Set the PortEntryStatus (specifying the location and port number) to valid (1). |
-

Command Format

```
SNMP_SET <card>PortEntryStatus.<location>.<PortNumber> underModification
```

```
SNMP_SET <card>PortState.<location>.<PortNumber> inactive
```

```
SNMP_SET <card>PortEntryStatus.<location>.<PortNumber> valid
```

Example

To deactivate the tenth DTG port at slot 1-1-3, use the following syntax to enter the command:

```
SNMP_SET dtgPortEntryStatus.3.10 integer 3
```

```
SNMP_SET dtgPortState.3.10 integer 2
```

```
SNMP_SET dtgPortEntryStatus.3.10 integer 1
```

The commands in this example perform the following functions:

- The first command sets the tenth DTG port to underModification. This indicates to other cooperating management applications that you are modifying the port.
- The second command sets the DTG port state to inactive.
- The third command sets the DTG port to valid. This command verifies the setting (inactive) and releases the port.

Modifying Port Parameters

You can set individual ports on an internal service circuit or network interface card to available or unavailable (activated or deactivated) with the PortState object. To complete this function, access the PortEntryStatus and the RLS hardware address. Then indicate the port(s) you are modifying.

The following special considerations affect the modification of port parameters:

- If you are setting a path for a port, first deactivate the port with the PortState object. While the port is deactivated, it cannot be allocated to a call. To make the port available for use, set the port to active with the PortState object.
- To reset a port, first deactivate the port, then reactivate it.
- NBCs and BRCs have no ports.
- When you deactivate a port, call processing no longer tries to assign any calls to the port. It is not necessary to remove the port from its resource group. This allows maintenance personnel to minimize the effects of a single port failure on the system.

To modify line card port parameters, enter an SNMP GET command to see all the read-write parameters. Decide which parameters you need to modify, and then perform the following steps:

-
- Step 1** Set the EntryStatus object (specifying the location and port number) to underModification (3).
- Step 2** Set the new parameter values (specifying the location and port number).
- Step 3** Set the EntryStatus object (specifying the location and port number) to valid (1).
- Step 4** To confirm that the parameter was changed, enter an SNMP_GET command.
-



Note

You can modify more than one parameter at the same time. Refer to the MIB Reference Guide for a list of attributes for each port.

Command Format

```
SNMP_SET <card>PortEntryStatus.<location>.<PortNumber> underModification
SNMP_SET <ObjectName>.<location>.<PortNumber> <type> <value>
SNMP_SET <card>PortEntryStatus.<location>.<PortNumber> valid
```

Example

To change the COS of the fifth PRI/N port at slot 1-1-7 to always off hook and originating, enter the following command syntax:

```
SNMP_SET isdnPortEntryStatus.7.5 integer 3
SNMP_SET isdnPortCos.7.5 integer 4
SNMP_SET isdnPortEntryStatus.7.5 integer 1
```

The commands in this example perform the following functions:

- The first command declares that you are modifying something in the ISDN card port table. This is an indication to other cooperating management applications that you are modifying the table entry.

- The second command declares that you are setting the value of `isdnPortCos` to `oa (4)`, or "Always Off-hook and Originating".
- The third command sets the value of `isdnPortEntryStatus` to `1`. This setting verifies the changes and releases the ISDN port object.

Error Status

If any of the operations described in the preceding sections fails, enter an `SNMP_GET` command on the `PortErrorStatus` object specifying the location and port number.

Command Format

```
SNMP_GET <card>PortErrorStatus.<location>.<PortNumber>
```

Example

```
SNMP_GET isdnPortErrorStatus.7.5
```

In the preceding example, the value returned tells you the error value. These values are defined in the individual `<card>PortErrorStatus` objects. Refer to the index to locate more information for individual `PortErrorStatus` objects.

Rule Configuration

Call processing requires various signaling and supervision actions, such as collecting or sending digits. Inpulse and outpulse rules define the sequence of these actions for each type of call that an application manages.

You can define up to 255 inpulse rules and 255 outpulse rules using specific tokens to designate each action within a rule. Up to 16 tokens are allowed for each inpulse or outpulse rule. Refer to the *Cisco VCO/4K System Administrator's Guide* for a list of the tokens and a description of how the tokens function.

You can perform the following functions with the objects from the inpulse and outpulse rule tables in the VCO/4K MIB:

- Add a rule
- Delete a rule
- Add tokens to a rule
- View rule parameters
- Modify rule parameters
- Get the error status

Adding a Rule

To add a rule, use the following steps:

-
- Step 1 Set EntryStatus (specifying the rule number) to createRequest (2).
 - Step 2 Set EntryStatus (specifying the rule number) to valid (1).
-

Command Format

```
SNMP_SET <rule>EntryStatus.<rulenumber> createRequest
SNMP_SET <rule>EntryStatus.<rulenumber> valid
```

Example

To add the fifth outpulse rule, use the following command syntax:

```
SNMP_SET outpulseRuleEntryStatus.5 integer 2
SNMP_SET outpulseRuleEntryStatus.5 integer 1
```

When you add a rule, all the parameters are set at the defaults.

Deleting a Rule

To delete a rule, use the following steps:

-
- Step 1 Set EntryStatus (specifying the rule number) to underModification (3).
 - Step 2 Set all the tokens to empty (0).
 - Step 3 Set the EntryStatus (specifying the rule number) to valid (1)
 - Step 4 Set the EntryStatus (specifying the rule number) to invalid (4).
-

Command Format

```
SNMP_SET <rule>EntryStatus.<rulenumber> underModification
SNMP_SET <rule>EntryStatus.<rulenumber>.<token-number> empty
SNMP_SET <rule>EntryStatus.<rulenumber> valid
SNMP_SET <rule>EntryStatus.<rulenumber> invalid
```

Example

To delete the inpulse rule with the index value of 2, use the following command syntax:

```
SNMP_SET inpulseRuleEntryStatus.2 integer 3
```

```
SNMP_SET inpulseRuleToken.2.1 integer 0
```

```
SNMP_SET inpulseRuleEntryStatus.2 integer 1
```

```
SNMP_SET inpulseRuleEntryStatus.2 integer 4
```

The commands in this example perform the following functions:

- The first command sets the EntryStatus for inpulse rule number 2 to underModification.
- The second command sets the inpulse rule token (rule number 2, token number 1) to empty. This step is repeated for each token in reverse order.
- The third command sets the EntryStatus for inpulse rule number 2 to valid.
- The fourth command sets the EntryStatus for inpulse rule number 2 to invalid. This command verifies the settings and deletes rule number 2.

Adding Tokens to a Rule

To add a token to an inpulse rule, use the following steps.

-
- | | |
|---------------|--|
| Step 1 | Set the EntryStatus for the rule (specifying the rule number) to underModification (3). This indicates to other cooperating management applications that you are modifying the rule. |
| Step 2 | Set the token (specifying the rule number and token number) to the desired value (0 to 47). Refer to Chapter 10, “Data Group,” for token values for the inpulseRuleToken object. |
| Step 3 | Set the EntryStatus (specifying the rule number to valid (1)). |
-

Command Format

```
SNMP_SET <rule>EntryStatus.<rulenum> underModification
```

```
SNMP_SET <rule>Token.<rulenum>.<token-number> <value>
```

```
SNMP_SET <rule>EntryStatus.<rulenum> valid
```

Example

To add a DTMF token to the fifth rule, use the following syntax to enter the command:

```
SNMP_SET inpulseRuleEntryStatus.5 integer 3
```

```
SNMP_SET inpulseRuleToken.5.1 integer 5
```

```
SNMP_SET inpulseRuleEntryStatus.5 integer 1
```

The commands in the preceding example have the following meanings:

- The first command sets the EntryStatus for inpulse rule number 5 to underModification. This command indicates to other cooperating management applications that you are modifying the rule.

- The second command assigns the value of 5 (dtmf) to the first impulse rule token for rule number 5.
- The third command sets the EntryStatus for impulse rule number 5 to valid. This command verifies the settings and releases the impulse rule.

Viewing Rule Parameters

To view the rule parameters, enter an `SNMP_GET` command. The following command shows the general syntax for viewing rule parameters. Refer to the *Cisco VCO/4K System Messages* document for a list of the errors.

Command Format

```
SNMP_GET <rule>Token.<rulenumber>.<token-number>
```

Example

To get the first token of rule 5, enter the following command syntax:

```
SNMP_GET impulseRuleToken.5.1
```

In this example, if you had deleted the impulse rule (or the rule was never configured) the value returned is an error. If the rule exists, the value returned is 5.

Modifying Rule Parameters

To modify the rule parameters, enter an `SNMP_GET` command to determine all the read-write parameters. Decide which parameters require modifying, and then perform the following steps:

Step 1 Set the EntryStatus object (specifying the rule number) to underModification (3).

Step 2 Set the new parameter values (specifying the rule number and token number).



Note You can set the value of more than one parameter at the same time.

Step 3 Set the EntryStatus object (specifying the rule number) to valid (1).

To confirm that the parameters were changed, enter an `SNMP_GET` command.

Command Format

```
SNMP_SET <rule>EntryStatus.<rulenumber> underModification
```

```
SNMP_SET <rule>Token.<rulenumber>.<token-number> <value>
```

```
SNMP_SET <rule>EntryStatus.<rulenumber> valid
```

Example

To modify the first token of output rule number 2, use the following syntax to enter the command:

```
SNMP_SET outputRuleEntryStatus.2 integer 3
```

```
SNMP_SET outputRuleToken.2.1 integer 5
```

```
SNMP_SET outputRuleEntryStatus.2 integer 1
```

The commands in this example perform the following functions:

- The first command declares that you are modifying output rule number 2. This indicates to other cooperating management applications that you are modifying the entry in this table.
- The second command declares that you are adding an MFCR2 outputRuleToken in position 1 to rule number 2.
- The third command declares that you are setting the value of outputRuleEntryStatus to 1. This setting verifies your changes and releases the rule.

Error Status

If any of the operations described in the preceding sections fails, enter an SNMP_GET command on the ErrorStatus object.

Command Format

```
SNMP_GET outputRuleErrorStatus.<rulenum>
```

Example

```
SNMP_GET outputRuleErrorStatus.2
```

In this example, the value returned tells you what the error is. These values are defined in the ErrorStatus object for each table in the MIB.

Resource Group Configuration

A resource group is a collection of functionally similar ports. Each resource group consists of a name, number, hunt type, and list of resources. You can define up to 999 resource groups in standard mode, and 1,920 resource groups in the VCO/4K's extended mode.

Using SNMP, you can assign ports to the resource group with objects in the VCO/4K MIB port tables (for example, lcResGroupIndex, tcResGroupIndex). The input to the port table object is the number of the resource group.

You can perform the following resource group functions using SNMP:

- Add a resource group
- Delete a resource group
- Modify resource group parameters
- View resource group parameters
- Add a port to a resource group

- Remove a port from a resource group
- View a port in a resource group
- Get the error status of a resource group

Consider the following caveats when working with resource groups:

- Internal service circuits must be grouped for the system to function properly. You must create one group for each circuit type. All circuits of the same type must be members of the same group.
- Network interface circuits should be grouped for optimum system performance and statistics collection. Network interface circuit groups and their members are application-dependent.
- The Resource Group Configuration Table lists the resource groups on the system, the ports that are assigned to them, and the spans to which the ports belong.
- Each port can belong to only one resource group. The card and port information must already be entered into the database before you can assign the port to a resource group. For information about adding and deleting cards from the database using the objects in the MIB, see Chapter 2, “System Configuration Group.”

Adding a Resource Group

When you add a new resource group, set the `resGroupIndex` object to `createRequest`, set the name and the parameters of the resource group, and then set the resource group parameters to valid. All of the MIB objects necessary to perform these functions are in the resource group table. To add a resource group, use the following sample steps.

-
- | | |
|---------------|--|
| Step 1 | Set the <code>resGroupEntryStatus</code> (specifying the resource group number) to <code>createRequest</code> (2). |
| Step 2 | Set the parameters of the resource group to a specified value. |
| Step 3 | Set the <code>resGroupEntryStatus</code> (specifying the resource group number) to <code>valid</code> (1). |
-

Command Format

```
SNMP_SET resGroupEntryStatus.<resGrpNumber> createRequest
SNMP_SET <ObjectName>.<resGrpNumber> <value>
SNMP_SET resGroupEntryStatus.<resGrpNumber> valid
```

Example

To add a resource group with the index value of 5, use the following syntax to enter the command:

```
SNMP_SET resGroupEntryStatus.5 integer 2
SNMP_SET resGroupName.5 string "Group1"
SNMP_SET resGroupEntryStatus.5 integer 1
```

The commands in this example perform the following functions:

- The first command sets the `EntryStatus` of the resource group with the index value of 5 to `createRequest`.

- The second command gives the new resource group the name Group1.
- The third command sets the EntryStatus of the resource group with the index value of 5 to valid. This command also releases the resource group.

Deleting a Resource Group

To delete a port from a resource group, set the EntryStatus object (specifying the resource group number) to invalid (4).

Command Format

```
SNMP_SET reGroupEntryStatus.<resGrpNumber> invalid
```

Example

To delete the resource group with the index value of 2, use the following command syntax:

```
SNMP_SET resGroupEntryStatus.2 integer 4
```

This command deletes the resource group with an index value of 2.

Modifying Resource Group Parameters

When you modify the resource group parameters, you choose how you want the resource group to function. You can modify the following resource group parameters:

- resGroupName
- resGroupBusyThres
- resGroupHuntType
- resGroupOwnerString
- resGroupRhuntThres
- resGroupEntrystatus

To modify the resource group parameters, enter an SNMP_GET command to determine all the read-write parameters. Decide which parameters require modifying, and then perform the following steps:

-
- | | |
|---------------|--|
| Step 1 | Set the EntryStatus object (specifying the resource group) to underModification (3). |
| Step 2 | Set the new parameter values (specifying the resource group and data type value). |
| Step 3 | Set the EntryStatus object (specifying the resource group) to valid (1). |
-

Command Format

```
SNMP_SET resGroupEntryStatus.<resGrpNumber> underModification
```

```
SNMP_SET <ObjectName>.<resGrpNumber> <value>
```

```
SNMP_SET resGroupEntryStatus.<resGrpNumber> valid
```

Example

```
SNMP_SET resGroupEntryStatus.5 integer 3
```

```
SNMP_SET resGroupHuntType.5 integer 1
```

```
SNMP_SET resGroupEntryStatus.5 integer 1
```

In this example, the commands listed above perform the following functions:

- The first command sets the resource group EntryStatus object in resource group number 5 to underModification. This indicates to other cooperating management stations that you are modifying the resource group.
- The second command sets the resGroupHuntType object to the data type value of 1 (rotary).
- The third command sets the resource group EntryStatus object in resource group number 5 to valid. This command verifies the setting and releases the resource group.

Viewing Resource Group Parameters

To view the resource group parameters, enter an SNMP_GET command.

Command Format

```
SNMP_GET <ObjectName>.<resourcegroup>
```

Example

To view the busy threshold of resource group 5, enter the following command syntax:

```
SNMP_GET resGroupBusyThres.5
```

In this example, if you had deleted the resource group (or the resource group was never configured) the value returned is an error. If the resource group exists, the value returned is 5. Refer to *Cisco VCO/4K System Messages* for a list of error messages.

Modifying Resource Group Port Parameters

To modify a resource group port, enter an SNMP_GET command to see all the read-write parameters. Decide which parameters need modifying, and then use the steps outlined in the following sections to add, delete, and view a port in resource groups.

Adding a Port to a Resource Group

Use the following steps to add a port to a resource group:

-
- Step 1** Set the resGroupEntryStatus object (specifying the resource group number) to underModification (3).
 - Step 2** Set the port EntryStatus (specifying the location and port number) to underModification (3).
 - Step 3** Set the portResGroupIndex (specifying the location and port number) to point to the resource group number to which it should belong.

- Step 4** Set the port EntryStatus (specifying the location and port number) to valid (1).
- Step 5** Set the resGroupEntryStatus (specifying the resource group number) to valid (1).
-

Command Format

```
SNMP_SET resGroupEntryStatus.<resGrpNumber> underModification
SNMP_SET <card>PortEntryStatus.<location>.<PortNumber> underModification
SNMP_SET <card>resGroupIndex.<location>.<PortNumber> integer <resGrpNumber>
SNMP_SET <card>PortEntryStatus.<location>.<PortNumber> valid
SNMP_SET resGroupEntryStatus.<resGrpNumber> valid
```

Example

To add the tenth port of the line card at slot 1-1-5 to resource group number 7, use the following syntax to enter the command:

```
SNMP_SET resGroupEntryStatus.7 integer 3
SNMP_SET lcPortEntryStatus.5.10 integer 3
SNMP_SET lcResGroupIndex.5.10 integer 7
SNMP_SET lcPortEntryStatus.5.10 integer 1
SNMP_SET resGroupEntryStatus.7 integer 1
```

The commands in this example perform the following functions:

- The first command sets the EntryStatus for resource group 7 to underModification.
- The second command sets the EntryStatus for the line card port to underModification.
- The third command sets the line card resource group index for resource group number 7.
- The fourth command sets the EntryStatus for the line card port to valid.
- The fifth command sets the EntryStatus for the resource group to valid. This command verifies the settings and releases the resource group.

Deleting a Port from a Resource Group

Use the following steps to remove a port from a resource group:

- Step 1** Set the resGroupEntryStatus object (specifying the resource group number) to underModification (3).
- Step 2** Set the port EntryStatus object (specifying the location and port number) to underModification (3).
- Step 3** Set the portResGroupIndex (specifying the location and port number) to empty (0).
- Step 4** Set the port EntryStatus (specifying the location and port number) to valid (1).
- Step 5** Set the resGroupEntryStatus (specifying the resource group number) to valid (1).
-

Command Format

```
SNMP_SET resGroupEntryStatus.<resGroupNumber> underModification
SNMP_SET <card>PortEntryStatus.<location>.<PortNumber> underModification
SNMP_SET <card>ResGroupIndex.<location>.<PortNumber> empty
SNMP_SET <card>PortEntryStatus.<location>.<PortNumber> valid
SNMP_SET resGroupEntryStatus.<resGroupNumber> valid
```

Example

To remove the tenth port of the line card at slot 1-15 in resource group number 7, use the following syntax to enter the command:

```
SNMP_SET resGroupEntryStatus.7 integer 3
SNMP_SET lcPortEntryStatus.5.10 integer 3
SNMP_SET lcResGroupIndex.5.10 integer 0
SNMP_SET lcPortEntryStatus.5.10 integer 1
SNMP_SET resGroupEntryStatus.7 integer 1
```

The commands in this example have the following meanings:

- The first command sets the EntryStatus for resource group 7 to underModification.
- The second command sets the EntryStatus for the line card (location 5) port (position 10) to underModification.
- The third command sets the line card resource group index for the port to empty.
- The fourth command sets the EntryStatus for the line card (index value of 5) port (position 10) to valid.
- The fifth command sets the EntryStatus for resource group 7 to valid. This command verifies the settings and releases the resource group.

Viewing a Port in a Resource Group

To view the ports in a resource group enter an SNMP_GET command on the resGroupPortTable.

Error Status

If any of the operations described in the preceding sections fails, enter an SNMP_GET command on the ErrorStatus object. Refer to the resGrpErrorStatus object for a list of error messages.

Command Format

```
SNMP_GET <resourcegroup>ErrorStatus.<resGrpNumber>
```

Example

SNMP_GET resGroupErrorStatus.7

In this example, the value returned is the numerical value for the error. These values are defined in the ErrorStatus object for each table in the MIB.

Message Templates

There are three kinds of message templates:

- Answer Supervision Templates
- ISDN Supervision Templates
- ISDN Message Templates

You can perform the following functions on the message templates:

- Add templates
- Delete templates
- Modify templates
- View template parameters
- Get the error status on templates

Answer Supervision Templates

Answer Supervision Templates are called from the WAIT SUP and FINAL SUP outpulse rule tokens. Each template is a set of system responses to the detection of specific signaling events. Signaling events include detection of call progress tones such as dial tone, ringback, busy tone, reorder, and special information (SIT) tones. Other recognized signaling events are voice detection, voice cessation, wink, true answer, grace timing, pager cue tones, and hookflash. For detailed information concerning the use of Answer Supervision Templates, refer to the *Cisco VCO/4K System Administrator's Guide*.

Adding an Answer Supervision Template

To add a new template, use the following steps:

-
- | | |
|--------|--|
| Step 1 | Set the ansSupTempEntryStatus to createRequest (2). |
| Step 2 | Set the object(s) you want to add to the Answer Supervision Template table to data-type value. |
| Step 3 | Set the ansSupTempEntryStatus to valid (1). |
-

Command Format

```
SNMP_SET ansSupTempEntryStatus.<TemplateName> createRequest
```

```
SNMP_SET ansSupTemp<ObjectName>.<TemplateName> <value>
```

```
SNMP_SET ansSupTemp<ObjectName>.<TemplateName> <value>
SNMP_SET ansSupTempEntryStatus.<TemplateName> valid
```

Example

```
SNMP_SET ansSupTempEntryStatus.5 integer 2
SNMP_SET ansSupTempDialTone.5 integer 1
SNMP_SET ansSupTempRingback.5 integer 2
SNMP_SET ansSupTempEntryStatus.5 integer 1
```

In this example, the commands perform the following functions:

- The first command sets the EntryStatus for answer supervision template number 5 to createRequest.
- The second command sets the DialTone object to ok.
- The third command sets the Ringback object to okrep.
- The fourth command sets the EntryStatus for answer supervision template number 5 to valid. This command verifies the settings, adds the template to the table and releases the template.

Deleting an Answer Supervision Template

To delete a template, use the following steps:

Step 1 Set the ansSupTempEntryStatus of the Answer Supervision Template you want to delete to underModification (3).

Step 2 Set all the object(s) in the Answer Supervision Template you want to delete to empty (0).



Note You must set all the objects to empty (0) to delete the template, then set the EntryStatus of the template to valid (1), and then invalid (4).

Step 3 Set the ansSupTempEntryStatus to valid (1).

Step 4 Set the ansSupTempEntryStatus to invalid (4).

Command Format

```
SNMP_SET ansSupTempEntryStatus.<TemplateName> underModification
SNMP_SET ansSupTemp<ObjectName>.<TemplateName> empty
SNMP_SET ansSupTemp<ObjectName>.<TemplateName> empty
SNMP_SET ansSupTempEntryStatus.<TemplateName> valid
SNMP_SET ansSupTempEntryStatus.<TemplateName> invalid
```

Example

```
SNMP_SET ansSupTempEntryStatus.5 integer 3
```

```
SNMP_SET ansSupTempDialTone.5 integer 0
SNMP_SET ansSupTempRingback.5 integer 0
SNMP_SET ansSupTempEntryStatus.5 integer 1
SNMP_SET ansSupTempEntryStatus.5 integer 4
```

In this example, the commands perform the following functions:

- The first command sets the EntryStatus object for answer supervision template number 5 to underModification. This indicates to other cooperating management stations that you are modifying this template.
- The second command sets the DialTone object to empty.
- The third command sets the Ringback object to empty.
- The fourth command sets the EntryStatus object to valid.
- The fifth command sets the EntryStatus object to invalid. This command verifies the settings, deletes, and then releases the template.

Modifying Answer Supervision Template Parameters

When you modify Answer Supervision Template parameters, you choose how you want the template to function.

To modify template parameters, enter an SNMP_GET command to see all the read-write parameters. Decide which parameters you need to modify, and then perform the following steps:

-
- Step 1** Set the ansSupTempEntryStatus to underModification (3).
- Step 2** Set the object(s) in the Answer Supervision Template table you want to modify to the desired data-type value.
- Step 3** Set the ansSupTempEntryStatus to valid (1).
-

Command Format

```
SNMP_SET ansSupTempEntryStatus.<TemplateName> underModification
SNMP_SET ansSupTemp<ObjectName>.<TemplateName> <value>
SNMP_SET ansSupTemp<ObjectName>.<TemplateName> <value>
SNMP_SET ansSupTempEntryStatus.<TemplateName> valid
```

Example

```
SNMP_SET ansSupTempEntryStatus.5 integer 3
SNMP_SET ansSupTempDialTone.5 integer 3
SNMP_SET ansSupTempRingback.5 integer 2
SNMP_SET ansSupTempEntryStatus.5 integer 1
```

In this example, the commands perform the following functions:

- The first command sets the EntryStatus object for answer supervision template number 5 to underModification. This indicates to other cooperating management stations that you are modifying this template.
- The second command sets the DialTone object to ansbk.
- The third command sets the Ringback object to okrep.
- The fourth command sets the EntryStatus object to valid. This verifies the settings and releases the template.

Viewing an Answer Supervision Template

To view parameters for Answer Supervision Template objects, enter an SNMP_GET command.

Command Format

```
SNMP_GET <ObjectName>.<TemplateName>
```

Example

```
SNMP_GET ansSupTempDialTone.5
SNMP_GET ansSupTempRingback.5
```

In this example, the parameters for the dial tone and ringback objects in Answer Supervision Template number 5 are displayed.

Error Status

If any of the operations described in the preceding sections fails, enter an SNMP_GET command on the ErrorStatus object. Refer to the ansSupTempErrorStatus object in Chapter 10, “Data Group,” for a list of the error messages.

Example

```
SNMP_GET ansSupTempErrorStatus.5
```

In this example, the value returned is the numerical value for the error. These values are defined in the ErrorStatus object for each table in the MIB.

ISDN Supervision Templates

Use the ISDN Supervision Templates to configure up to 24 supervision templates for use with ISDN SUP [xx] outpulse rule token. Each template defines a set of system responses to the detection of specific ISDN messages such as alerting, connect, progress, and callproc (call proceeding).

Calls are not marked stable when a template is being executed unless the template specifies to do so. If the rule ends and the call is not already stable, receipt of the connect message transitions the call to stable.

Adding an ISDN Supervision Template

To add a new template, use the following steps:

-
- | | |
|---------------|--|
| Step 1 | Set the isdnSupTempEntryStatus to createRequest (2). |
| Step 2 | Set the object(s) you want to add to the ISDN Supervision Template table to the desired data-type value. |
| Step 3 | Set the isdnSupTempEntryStatus to valid (1). |
-

Command Format

```
SNMP_SET isdnSupTempEntryStatus.<TemplateName> createRequest
SNMP_SET isdnSupTemp<ObjectName>.<TemplateName> <value>
SNMP_SET isdnSupTemp<ObjectName>.<TemplateName> <value>
SNMP_SET isdnSupTempEntryStatus.<TemplateName> valid
```

Example

```
SNMP_SET isdnSupTempEntryStatus.5 integer 2
SNMP_SET isdnSupTempTime.5 integer 1
SNMP_SET isdnSupTempAlerting.5 integer 2
SNMP_SET isdnSupTempEntryStatus.5 integer 1
```

In this example, the commands perform the following functions:

- The first command sets the EntryStatus object for ISDN supervision template number 5 to createRequest.
- The second command sets the Time object in the template to ok.
- The third command sets the Alerting object in the template to okrep.
- The fourth command sets the EntryStatus object to valid. This command verifies the settings and releases the template.

Deleting an ISDN Supervision Template

To delete a template, use the following steps:

-
- Step 1 Set the `isdnSupTempEntryStatus` of the ISDN Supervision Template you want to delete to `underModification` (3).
 - Step 2 Set all the object(s) from the ISDN Supervision Template you want to delete to empty (0).



Note You must set all the objects to empty (0), then set the template to valid (1), and then invalid (4) before you can delete the entire template.

- Step 3 Set the `isdnSupTempEntryStatus` to `valid` (1).
 - Step 4 Set the `isdnSupTempEntryStatus` to `invalid` (4).
-

Command Format

```
SNMP_SET isdnSupTempEntryStatus.<TemplateName> underModification
SNMP_SET isdnSupTemp<ObjectName>.<TemplateName> empty
SNMP_SET isdnSupTemp<ObjectName>.<TemplateName> empty
SNMP_SET isdnSupTempEntryStatus.<TemplateName> valid
SNMP_SET isdnSupTempEntryStatus.<TemplateName> invalid
```

Example

```
SNMP_SET isdnSupTempEntryStatus.5 integer 3
SNMP_SET isdnSupTempTime.5 integer 0
SNMP_SET isdnSupTempAlerting.5 integer 0
SNMP_SET isdnSupTempEntryStatus.5 integer 1
SNMP_SET isdnSupTempEntryStatus.5 integer 4
```

In this example, the commands perform the following functions:

- The first command sets the `EntryStatus` object for ISDN supervision template number 5 to `underModification`. This command indicates to other cooperating management stations that you are modifying this template.
- The second command sets the `Time` object to empty.
- The third command sets the `Alerting` object to empty.
- The fourth command sets the `EntryStatus` object to `valid`.
- The fifth command sets the `EntryStatus` object to `invalid`. This command deletes the template.

Modifying ISDN Supervision Template Parameters

When you modify the ISDN Supervision Template parameters, you choose how you want the template to function.

To modify a template, use the following steps:

-
- | | |
|---------------|---|
| Step 1 | Set the <code>isdnSupTempEntryStatus</code> to <code>underModification</code> (3). |
| Step 2 | Set the object(s) in the ISDN Supervision Template table you want to modify to the desired data-type value. Refer to Chapter 6, “Single-Span ISDN Card Group,” for a list of possible values for each object. |
| Step 3 | Set the <code>isdnSupTempEntryStatus</code> to <code>valid</code> (1). |
-

Command Format

```
SNMP_SET isdnSupTempEntryStatus.<TemplateName> underModification
SNMP_SET isdnSupTemp<ObjectName>.<TemplateName> <value>
SNMP_SET isdnSupTemp<ObjectName>.<TemplateName> <value>
SNMP_SET isdnSupTempEntryStatus.<TemplateName> valid
```

Example

```
SNMP_SET isdnSupTempEntryStatus.5 integer 3
SNMP_SET isdnSupTempTime.5 integer 3
SNMP_SET isdnSupTempAlerting.5 integer 4
SNMP_SET isdnSupTempEntryStatus.5 integer 1
```

In this example, the commands perform the following functions:

- The first command sets the `EntryStatus` object for the ISDN supervision template number 5 to `underModification`. This indicates to other cooperating management stations that you are modifying this template.
- The second command sets the `Time` object to `ansbk`.
- The third command sets the `Alerting` object to `ansrep`.
- The fourth command sets the `EntryStatus` object to `valid`. This command verifies the settings and releases the template.

Viewing an ISDN Supervision Template

Command Format

```
SNMP_GET <ObjectName>.<TemplateName>
```

Example

```
SNMP_GET isdnSupTempDialTone.5
```

In this example, the status of the dial tone object in ISDN Supervision Template number 5 is displayed.

Error Status

If any of the operations described in the preceding sections fail, enter an `SNMP_GET` command on the `ErrorStatus` object. Refer to the `isdnSupTempErrorStatus` object in Chapter 6, “Single-Span ISDN Card Group,” for a list of the error messages.

Example

```
SNMP_GET isdnSupTempErrorStatus.5
```

In this example, the value returned is the numerical value for the error. These values are defined in the `ErrorStatus` object for each table in the MIB.

ISDN Message Templates

Up to 96 message templates can be defined for use in rule processing of ISDN calls. Template processing is reported to the host through the ISDN Inpulse Rule Complete and ISDN Port Change of State reports. (For more information concerning these reports, refer to the *Cisco VCO/4K Extended Programming Reference*.) When reporting is done in conjunction with inpulse or outpulse rule processing, timing of the reports is controlled by the reporting control tokens contained in the rule, or by the supervision template.

Adding an ISDN Message Template

To add a new template, use the following steps:

-
- Step 1 Set the `isdnMessageTempEntryStatus` to `createRequest` (2).
 - Step 2 Set the `isdnMessageTempMessage` to `Alerting` (2).
 - Step 3 Set the `isdnMessageTempRT` to `Transmit` (2).
 - Step 4 Set the `isdnTempAction` to `IE` (5).
 - Step 5 Set the `isdnTempIeData` to `Bearer` (1).
 - Step 6 Set the `isdnMessageTempEntryStatus` to `valid` (1).
-

Command Format

```
SNMP_SET isdnMessageTempEntryStatus.<TemplateName> createRequest
```

```
SNMP_SET isdnMessageTempMessage.<TemplateName> <value>
```

```
SNMP_SET isdnMessageTempRT.<TemplateName> <value>
```

```
SNMP_SET isdnTempAction.<TemplateName>.1 <value>
SNMP_SET isdnTempIeData.<TemplateName>.1 <value>
SNMP_SET isdnMessageTempEntryStatus.<TemplateName> valid
```

Example

```
SNMP_SET isdnMessageTempEntryStatus.5 integer 2
SNMP_SET isdnMessageTempMessage.5 integer 2
SNMP_SET isdnMessageTempRT.5 integer 2
SNMP_SET isdnTempAction.5.1 integer 5
SNMP_SET isdnTempIeData.5.1 integer 1
SNMP_SET isdnMessageTempEntryStatus.5 integer 1
```

In this example, the commands perform the following functions:

- The first command sets the EntryStatus object for ISDN message template number 5 to createRequest.
- The second command sets the Message object to alerting.
- The third command sets the RT object to t (transmit).
- The fourth command sets the TempAction object to ie (construct an IE for D-Channel transmit).
- The fifth command sets the TempIeData object to bearer.
- The sixth command sets the EntryStatus object to valid. This verifies the settings and releases the template.

Deleting an ISDN Message Template

To delete an ISDN Message Template, use the following steps:

-
- Step 1** Set the isdnMessageTempEntryStatus to underModification (3).
- Step 2** Set all the isdnTempAction objects to empty (0).



Note It is necessary to set all the objects to empty (0) to delete them, then set the template to valid (1), and then invalid (4).

- Step 3** Set the isdnMessageTempRT to empty (0).
- Step 4** Set the isdnMessageTempMessage to empty (0).
- Step 5** Set the isdnMessageTempEntryStatus to valid (1).
- Step 6** Set the isdnMessageTempEntryStatus to invalid (4).
-

Command Format

```
SNMP_SET isdnMessageTempEntryStatus.<TemplateName> underModification
SNMP_SET isdnTempAction.<TemplateName(N to 1)> empty
SNMP_SET isdnMessageTempRT.<TemplateName> empty
SNMP_SET isdnMessageTempMessage.<TemplateName> empty
SNMP_SET isdnMessageTempEntryStatus.<TemplateName> valid
SNMP_SET isdnMessageTempEntryStatus.<TemplateName> invalid
```

Example

```
SNMP_SET isdnMessageTempEntryStatus.5 integer 3
SNMP_SET isdnTempAction.<TemplateName(1 to N)> integer 0
SNMP_SET isdnMessageTempRT.5 integer 0
SNMP_SET isdnMessageTempMessage.5 integer 0
SNMP_SET isdnMessageTempEntryStatus.5 integer 1
SNMP_SET isdnMessageTempEntryStatus.5 integer 4
```

In this example, the commands perform the following functions:

- The first command sets the EntryStatus for ISDN message template number 5 to underModification. This indicates to other cooperating management stations that you are modifying this template.
- The second command sets the Action object(s) to empty.
- The third command sets the RT object to empty.
- The fourth command sets the Message object to empty.
- The fifth command sets the EntryStatus object to valid.
- The sixth command sets the Entry Status object to invalid. The command verifies the settings and deletes the template.

Modifying ISDN Message Template Parameters

When you modify the ISDN Message Template parameters, you choose how you want the template to function. You can modify the following parameters:

- isdnMessageTempMessage
- isdnMessageTempEntryStatus
- isdnMessageTempRT
- isdnTempAction
- isdnMessageTempOwnerString
- isdnTempIeData

To modify a template, use the following steps:

-
- Step 1** Set the `isdnMessageTempEntryStatus` to `underModification` (3).
- Step 2** Set the object(s) in the ISDN Message Template table you want to modify to the desired data-type value.
- Step 3** Set the `isdnMessageTempEntryStatus` to `valid` (1).
-

Command Format

```
SNMP_SET isdnMessageTempEntryStatus.<TemplateName> underModification
SNMP_SET isdnMessageTemp<ObjectName>.<TemplateName> <value>
SNMP_SET isdnMessageTemp<ObjectName>.<TemplateName> <value>
SNMP_SET isdnMessageTempEntryStatus.<TemplateName> valid
```

Example

```
SNMP_SET isdnMessageTempEntryStatus.5 integer 3
SNMP_SET isdnMessageTempMessage.5 integer 3
SNMP_SET isdnMessageTempRT.5 integer 2
SNMP_SET isdnTempAction.5.1 integer 6
SNMP_SET isdnTempIeData.5.1 integer 2
SNMP_SET isdnMessageTempEntryStatus.5 integer 1
```

In this example, the commands have the following meanings:

- The first command sets the `EntryStatus` object of ISDN message template number 5 to `underModification`. This indicates to other cooperating management stations that you are modifying this template.
- The second command sets the `Message` object to `callproc` (call proceeding).
- The third command sets the `RT` object to `t` (transmit).
- The fourth command sets the `Action` object to `data` (specify hexadecimal bytes for D Channel transmit).
- The fifth command sets the `IeData` object to `cause`.
- The sixth command sets the `EntryStatus` object to `valid`. This command verifies the settings and releases the template.

Viewing an ISDN Message Template

Command Format

```
SNMP_GET <ObjectName>.<TemplateName>
```

Example

```
SNMP_GET isdnMessageTempEntryStatus.5
```

In this example, the EntryStatus for ISDN Message Template number 5 is displayed.

Error Status

If any of the operations described in the preceding sections fail, enter an SNMP_GET command on the ErrorStatus object. Refer to the isdnMessageTempErrorStatus object in Chapter 6, “Single-Span ISDN Card Group,” for a list of the error messages.

Example

```
SNMP_GET isdnMessageTempErrorStatus.5
```

In this example, the value returned is the numerical value for the error. These values are defined in the ErrorStatus object for each table in the MIB.

Routing Tables

The Routing Tables define call routes (resource group hunted and impulse/output rule executed) based on pattern matching of the dialed digits. Up to 1000 routes can be defined and distributed among the ten route tables. The routes do not have to be divided equally among the route tables (for example, one table can have 217 routes, another 105 routes, and so forth); users can also assign all 1000 possible routes to a single routing table.

Three exception routes (Exception Routing Table) are dedicated to handling the following conditions:

- Fewer than the minimum number of digits received
- No digits received
- No match found—invalid digits or table problems

A fourth route is dedicated to direct routing without pattern matching.

When a route is added, modified, or deleted, all matching or routing information is sorted and ordered numerically according to the pattern. For more information concerning routing tables, refer to the *Cisco VCO/4K TeleRouter Reference Guide*.

You can perform the following routing functions using SNMP:

- Add a routing table
- Add a route to a routing table
- Delete a routing table
- Modify the routing table parameters
- View the routing table parameters
- Get the error status

Adding a Routing Table

To add a new routing table, use the following steps:

-
- Step 1 Set the routeGrpEntryStatus of a particular routeGrpIndex to createRequest (2).
 - Step 2 Set the routeGrpTableName (1 to 12 alphanumeric characters).
 - Step 3 Set the routeGrpTemplateSize (from 1 to 20).
 - Step 4 Set the routeGrpTemplateStart position (from 1 to 20).
 - Step 5 Set the routeGrpMinimumDigits for pattern matching (from 0 to 20).
 - Step 6 Set the routeGrpEntryStatus to valid (1).
-

Command Format

```
SNMP_SET routeGrpEntryStatus.<RouteGrpNumber> createRequest
SNMP_SET routeGrpTableName.<RouteGrpNumber> string
SNMP_SET routeGrpTemplateSize.<RouteGrpNumber> integer
SNMP_SET routeGrpTemplateStart.<RouteGrpNumber> integer
SNMP_SET routeGrpMinimumDigits.<RouteGrpNumber> integer
SNMP_SET routeGrpEntryStatus.<RouteGrpNumber> valid
```

Example

```
SNMP_SET routeGrpEntryStatus.3 integer 2
SNMP_SET routeGrpTableName.3 string "Example"
SNMP_SET routeGrpTemplateSize.3 integer 4
SNMP_SET routeGrpTemplateStart.3 integer 4
SNMP_SET routeGrpMinimumDigits.3 integer 7
SNMP_SET routeGrpEntryStatus.3 integer 1
```

In this example, the commands perform the following functions:

- The first command creates a new route group entry.
- The second command assigns a name to the route group.
- The third command determines the number of consecutive digits used in pattern matching for the table.
- The fourth command determines the starting position of the digits used in pattern matching for the table.



Note This value must be less than or equal to the difference between the template size and 20, plus 1.

- The fifth command determines the number of digits the switch must detect when it uses pattern matching in the table.
- The sixth command sets the EntryStatus to valid. This command verifies the settings and releases the table.

Adding a Route to a Routing Table

When you add a new route to a table, the system sorts all the entries numerically according to the pattern specified. To add a new route to a table, use the following steps.



Note

You must add routes in sequence.

-
- Step 1 Set the routeGrpEntryStatus to underModification (3).
 - Step 2 Set the routeAdd to a string value.
 - Step 3 Set the routePrimaryGroup to the resource group number (0 to 63).
 - Step 4 Set the routePrimaryRule to the inpulse/outpulse rule number (0 to 30).
 - Step 5 Set the routeSecondaryGroup to the resource group number (0 to 63).
 - Step 6 Set the routeSecondaryRule to the inpulse/outpulse rule number (0 to 30).
 - Step 7 Set the routeFinalGroup to the resource group number (0 to 63).
 - Step 8 Set the routeFinalRule to the inpulse/outpulse rule number (0 to 30).
 - Step 9 Set the routeGrpEntryStatus to valid (1).
-

Command Format

```
SNMP_SET routeGrpEntryStatus.<RouteGrpNumber> underModification
SNMP_SET routeAdd.<RouteGrpNumber>.<RouteNumber> string
SNMP_SET routePrimaryGroup.<RouteGrpNumber>.<RouteNumber> <value>
SNMP_SET routePrimaryRule.<RouteGrpNumber>.<RouteNumber> <value>
SNMP_SET routeSecondaryGroup.<RouteGrpNumber>.<RouteNumber> <value>
SNMP_SET routeSecondaryRule.<RouteGrpNumber>.<RouteNumber> <value>
SNMP_SET routeFinalGroup.<RouteGrpNumber>.<RouteNumber> <value>
SNMP_SET routeFinalRule.<RouteGrpNumber>.<RouteNumber> <value>
SNMP_SET routeGrpEntryStatus.<RouteGrpNumber> valid
```


Example

```
SNMP_SET routeGrpEntryStatus.3 integer 3
SNMP_SET routeAdd.3.5 string "555"
SNMP_SET routePrimaryGroup.3.5 integer 6
SNMP_SET routePrimaryRule.3.5 integer 8
SNMP_SET routeSecondaryGroup.3.5 integer 8
SNMP_SET routeSecondaryRule.3.5 integer 8
SNMP_SET routeFinalGroup.3.5 integer 8
SNMP_SET routeFinalRule.3.5 integer 8
SNMP_SET routeGrpEntryStatus.3 integer 1
```

In this example, the commands perform the following functions:

- The first command sets the routeGrpEntryStatus to underModification so a new route can be added.
- The second command adds a new route (5) to the routing group and sets the route pattern to 555.
- The third command sets the Primary group of route number 5 to route group number 6.
- The fourth through eighth commands set resource group numbers and inpulse/outpulse rule numbers associated with the particular route (5).
- The ninth command sets the resGrpEntryStatus to valid.

Deleting a Route from a Routing Table

To delete a route from a routing table, use the following steps:

-
- Step 1** Set the routeGrpEntryStatus of the Routing Table you want to delete to underModification (3).
- Step 2** Set the routeDelete to the valid route number.



Note You must delete all the routes in reverse order.

- Step 3** Set the routeGrpEntryStatus to valid (1).
-

Command Format

```
SNMP_SET routeGrpEntryStatus.<RouteGrpNumber> underModification
SNMP_SET routeDelete.<RouteGrpNumber>.<RouteNumber> integer
SNMP_SET routeGrpEntryStatus.<RouteGrpNumber> valid
```

Example

```
SNMP_SET routeGrpEntryStatus.5 integer 3
```

```
SNMP_SET routeDelete.5 integer 1
```

```
SNMP_SET routeGrpEntryStatus.5 integer 1
```

In this example, the commands perform the following functions:

- The first command sets the routeGrpEntryStatus for route group number 5 to underModification.
- The second command deletes route number 1 (a single route) from route group 5.
- The third command sets the routeGrpEntryStatus for route group 5 to valid.

Deleting a Routing Table

To delete a routing table, use the following steps:

Step 1 Set the routeGrpEntryStatus of the Routing Table you want to delete to underModification (3).

Step 2 Set the routeDelete to the valid route number.



Note You must delete all the routes in reverse order.

Step 3 Set the routeGrpEntryStatus to valid (1).

Step 4 Set the routeGrpEntryStatus to invalid (4).

Command Format

```
SNMP_SET routeGrpEntryStatus.<RouteGrpNumber> underModification
```

```
SNMP_SET routeDelete.<RouteGrpNumber>.<RouteNumber> integer
```

```
SNMP_SET routeGrpEntryStatus.<RouteGrpNumber> valid
```

```
SNMP_SET routeGrpEntryStatus.<RouteGrpNumber> invalid
```

Example

```
SNMP_SET routeGrpEntryStatus.5 integer 3
```

```
SNMP_SET routeDelete.5.1 integer 1
```

```
SNMP_SET routeGrpEntryStatus.5 integer 1
```

```
SNMP_SET routeGrpEntryStatus.5 Integer 4
```

In this example, the commands perform the following functions:

- The first command sets the routeGrpEntryStatus for route group number 5 to underModification.
- The second command deletes route number 1 from route group 5 (deletes all the routes in the route group).

- The third command sets the routeGrpEntryStatus for route group 5 to valid.
- The fourth command sets the routeGrpEntryStatus for route group 5 to invalid. This command deletes route group 5.

Modifying Routing Table Parameters

When you modify the Routing Table parameters, you choose how you want the table to function. You can modify the following parameters:

- exrteFinalGroup
- exrteFinalRule
- exrtePrimaryGroup
- exrtePrimaryRule
- exrteSecondaryGroup
- exrteSecondaryRule
- routeAdd
- routeDelete
- routeFinalGroup
- routeFinalRule
- routeGrpEntryStatus
- routeGrpMinimumDigits
- routeGrpOwnerString
- routeGrpTableName
- routeGrpTemplateSize
- routeGrpTemplateStart
- routeGrpTest
- routePrimaryGroup
- routePrimaryRule
- routeSecondaryGroup
- routeSecondaryRule

To modify a routing table, enter an SNMP_GET Command to see all the read-write parameters. Decide which parameters you need to modify, and then perform the following steps:

-
- Step 1** Set the routeGrpEntryStatus to underModification (3).
- Step 2** Set the object(s) in the Routing Group Table, Exception Routing Table, or Route Table you want to modify to the desired data-type value.
- Step 3** Set the routeGrpEntryStatus to valid (1).
-

Command Format

```
SNMP_SET routeGrpEntryStatus.<RouteGrpNumber> underModification
SNMP_SET route<ObjectName>.<RouteGrpNumber>.<RouteNumber> <value>
SNMP_SET route<ObjectName>.<RouteGrpNumber>.<RouteNumber> <value>
SNMP_SET routeGrpEntryStatus.<RouteGrpNumber> valid
```

Example

```
SNMP_SET routeGrpEntryStatus.3 integer 3
SNMP_SET routePrimaryGroup.3.1 integer 6
SNMP_SET routePrimaryRule.3.1 integer 6
SNMP_SET routeGrpEntryStatus.3 1
```

In this example, the commands perform the following functions:

- The first command sets the EntryStatus of the route group to underModification.
- The second command sets the value for routePrimaryGroup to 6. This indicates the resource group from which the switch should hunt for an outgoing port if the dialed digits match the corresponding pattern.
- The third command sets the value for routePrimaryRule to 6. This indicates the impulse/outpulse rule that the switch should execute if the dialed digits match the corresponding pattern.
- The fourth command sets the EntryStatus of the route group to valid.

Viewing the Routing Table Parameters

To view the Routing Table parameters, enter an SNMP_GET command. The following subsections show the sample commands for viewing the parameters for routing tables, for route groups, routes, and exception routes.

Viewing Route Group Parameters

Command Format

```
SNMP_GET routeGrp<ObjectName>.<RouteGrpNumber>
```

Example

```
SNMP_GET routeGrpMinimumDigits.3
```

This example displays the parameters for the MinimumDigits object.

Viewing Route Parameters

Command Format

```
SNMP_GET route<ObjectName>.<RouteGrpNumber>.<RouteNumber>
```

Example

```
SNMP_GET routePrimaryGroup.3.1
```

This example displays the parameters for the PrimaryGroup object.

Viewing Exception Route Parameters

Command Format

```
SNMP_GET exrte<ObjectName>.<RouteGrpNumber>.<RouteNumber>
```

Example

```
SNMP_GET exrteSecondaryRule.3.2
```

This example displays the parameters for the SecondaryRule object.

Error Status

If any of the operations described in the preceding sections fails, enter an SNMP_GET command on the ErrorStatus object. Refer to the routeGrpErrorStatus object in Chapter 10, “Data Group,” for a list of the error messages.

The following example performs an SNMP_GET request on Routing Group Table 3.

Example

```
SNMP_GET routeGrpErrorStatus.3
```

In this example, the value returned is the numerical value for the error in route group number 3. These values are defined in the ErrorStatus object for each table in the MIB.

