

Enable Four Bundles on a RAVE 188s/s-24

A tutorial

Introduction

This tutorial will present one method of accessing 2 bundles of audio, in both transmit and receive directions, via CobraNet™ with the RAVE™ 188s-24 models. This tutorial is also applicable to the RAVE 188s and 88s models.

The RAVE 188s-24 provides 8 channels of physical audio input and output. The 188s-24 ships from the factory with all 8 audio inputs destined for one outbound CobraNet bundle. Likewise, an inbound bundle is disassembled and delivered to the unit's 8 audio outputs. However, the RAVE 188s-24 is capable of processing up to 2 inbound and 2 outbound bundles. This additional accessibility requires software configuration using a Simple Network Management Protocol (SNMP) browser or related application. Once configured, all parameters may be retained in permanent memory within the RAVE device. This “persistence” of user-configurable settings prevents the loss of communications or bundle processing should there be a power cycle or reset event.

** It may be necessary to refer to additional on-line information if unfamiliar with SNMP. The QSC Audio website provides some basic configuration information and a detailed tutorial for getting started with SNMP for use with the RAVE products.*

The Network Example

The example given in this document consists of a 3-unit cluster in which two RAVEs bi-directionally communicate with a third. We will call the third unit the primary RAVE. We will label the two secondary RAVE 188s-24 units as devices “B” and “C” and the primary RAVE unit as device “A”. The custom configuration really only applies to device “A” in this example. However, each unit may be configured as required for a particular application.

In Figure 1 below, all audio channels are indicated with short solid arrows and all CobraNet bundles are indicated with longer dashed arrows. Audio channel inputs, shown pointing into the RAVEs, will be grouped into outbound bundles and delivered over the network. Inbound bundles will be received from the network, disassembled and then

delivered to the RAVE's audio outputs. Audio outputs are shown pointing away from the RAVE 188s-24 units. All audio channels associated with a bundle, as well as the bundle direction, are shown in color-coded groups. Above each audio channel group is a 4-digit alphanumeric value beginning with the letter B. The value indicates the bundle value that the associated channels will use for delivery over the CobraNet network. In this example, we have assumed that unicast delivery is a requirement of the network design. As each application differs, multicast delivery may be required for a given installation. If multicast delivery were used, there would be no need to alter the configuration of the transmitters at device A.

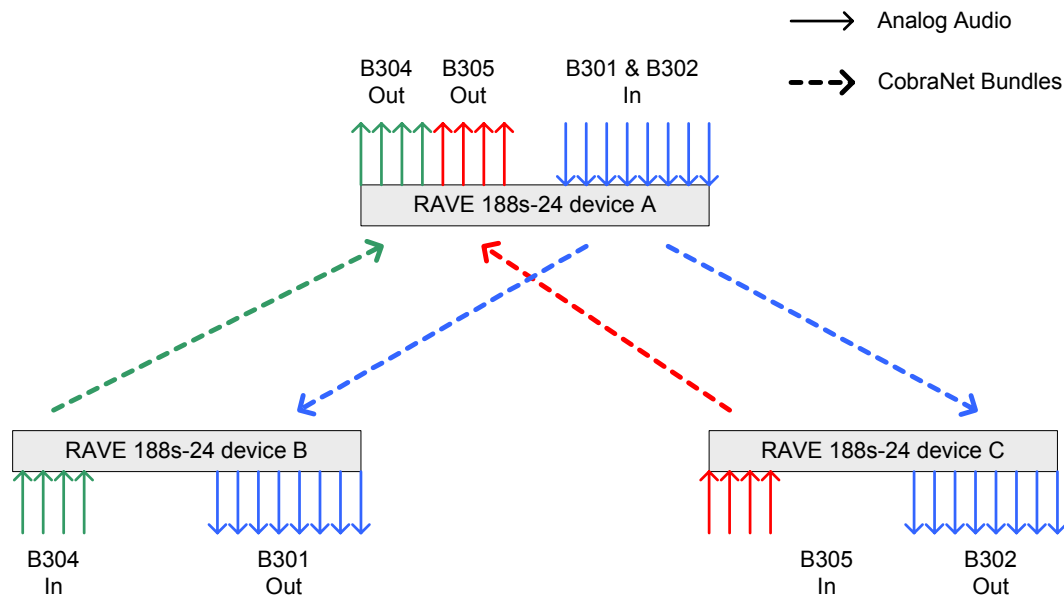


Figure 1

In Figure 1, eight channels of audio are present at the primary RAVE's inputs. These eight audio channels are copied and split into two groups. Each group is then packaged into a separate bundle and delivered over the network. The first group is packaged into unicast bundle 301 and delivered over the network to RAVE device B. The second group is packaged into unicast bundle 302 and delivered over the network to RAVE device C.

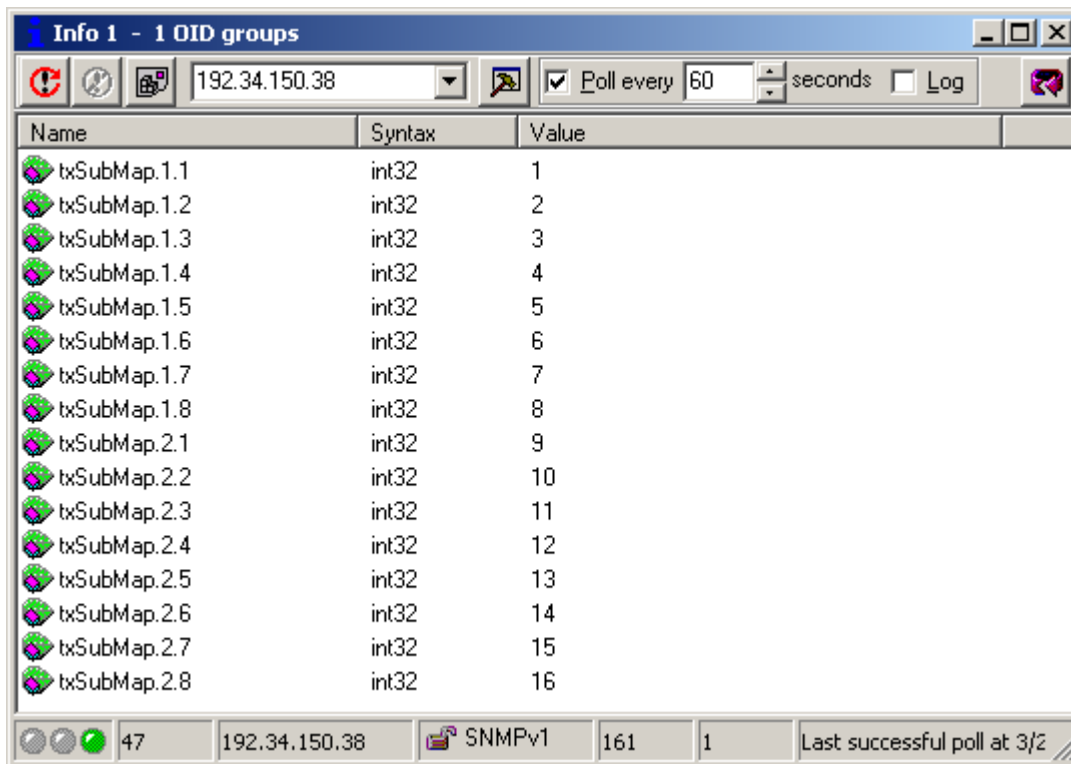
RAVE device B has four audio channels present at its inputs. These four channels are packaged into unicast bundle 304 and delivered over the network to the primary RAVE. RAVE device A receives these four channels and brings them out to the left four audio channel outputs.

RAVE device C also has four audio channels present at its inputs. These four channels are packaged into unicast bundle 305 and delivered over the network to the primary RAVE. RAVE device A receives these four channels and brings them out to the right four audio channel outputs, as shown in the figure.

Configuring the Transmitter on the primary RAVE

The configuration in Figure 1 requires duplicating the primary RAVE's audio inputs onto two separate bundles. This is fairly straightforward. This simply requires assigning each of the eight audio input channels, as mapped to locations 1 through 8 on the first bundle, to the management interface (MI) **txSubMap** variables 1 through 8 on the RAVE's second bundle. This sounds more confusing than it is. If we open the **txSubMap** leaf in an SNMP browser, we will see that there are sixteen variables available. Refer to Figure 2. Variables **txSubMap.1.n** are assigned to the first bundle and variables **txSubMap.2.n** are assigned to the second bundle. Each audio channel number replaces the "n" designator. Note, in this tutorial we have assigned the first bundle unicast value 301 and the second bundle unicast value 302.

Notice that there are 16 audio channels shown in Figure 2 but only eight audio channel inputs available on the RAVE 188s-24. Channels 9 through 16 simply don't exist on the RAVE 188s-24. In its default state, the RAVE 188s-24 normally assigns its eight audio channel inputs to the first group of **txSubMap** variables and simply disregards the second group. The point of this tutorial is to enable the second group of variables.



Name	Syntax	Value
txSubMap.1.1	int32	1
txSubMap.1.2	int32	2
txSubMap.1.3	int32	3
txSubMap.1.4	int32	4
txSubMap.1.5	int32	5
txSubMap.1.6	int32	6
txSubMap.1.7	int32	7
txSubMap.1.8	int32	8
txSubMap.2.1	int32	9
txSubMap.2.2	int32	10
txSubMap.2.3	int32	11
txSubMap.2.4	int32	12
txSubMap.2.5	int32	13
txSubMap.2.6	int32	14
txSubMap.2.7	int32	15
txSubMap.2.8	int32	16

Figure 2

As mentioned earlier, we are going to copy all eight input channels onto the second bundle.

To configure the transmitter on the primary RAVE 188s-24, we open the **transmitters** folder in the peakAudio MIB (Management Information Base) within our SNMP application. If using an SNMP browser with manual OID (object identifier) entry, the **transmitters** folder is assigned OID 2680.1.1.7. We have eliminated the preceding OID (usually 1.3.6.1.4.1.) throughout this document. A review of the MIB tree indicates that the value 2680 is the beginning of the peakAudio MIB and the **transmitters** folder is the 7th folder down from the **core Manager**.

We continue to open the **txSubChannelTable** and **txSubChannelEntry** subfolders under **transmitters**. We then select the **txSubMap** leaf. This leaf exposes the 16 channel default assignments (values 1 through 16) given to audio groups 1 and 2, as shown in Figure 2. Here, we copy audio channels 1 through 8 assigned to bundle 1 into the second **txSubMap** group for bundle 2. The values 1 through 8 should now be present on both **txSubMap** groups. The result is shown in Figure 3.

Name	Syntax	Value
txSubMap.1.1	int32	1
txSubMap.1.2	int32	2
txSubMap.1.3	int32	3
txSubMap.1.4	int32	4
txSubMap.1.5	int32	5
txSubMap.1.6	int32	6
txSubMap.1.7	int32	7
txSubMap.1.8	int32	8
txSubMap.2.1	int32	1
txSubMap.2.2	int32	2
txSubMap.2.3	int32	3
txSubMap.2.4	int32	4
txSubMap.2.5	int32	5
txSubMap.2.6	int32	6
txSubMap.2.7	int32	7
txSubMap.2.8	int32	8

Figure 3

Configuring the Receiver on the primary RAVE

On the receiving end, the primary RAVE device must receive two independent bundles from separate locations on the network and route them to its 8 audio outputs (Phoenix connectors). This can be done by assigning audio receive channels 1 through 4, on bundle 304, to the RAVE’s first four physical audio outputs (channels 1 through 4). The

remaining audio channels (channels 5 through 8) on bundle 304 are discarded. We then assign audio channels 1 through 4 on bundle 305 to the primary RAVE's last four physical audio outputs (channels 5 through 8). Again, the remaining audio channels (5 through 8) on bundle 305 are discarded. Refer again to Figure 1 for an overview of the channel routing.

Configuring the receive end is conceptually the same as the previous section, but it tends to confuse folks more. This is because receive channels are assigned values 33 through 48. The reason for the awkward values has to do with the CobraNet audio map. The audio map consists of 64 channels. Channels 1 through 32 are assigned to transmitters (physical inputs) and channels 33 through 64 are assigned to receivers (physical outputs). The RAVE supports 16 channels in both directions. Therefore, the RAVE transmitters are assigned channels 1 through 16 and the receivers are assigned channels 33 through 48.

In Figure 4, the RAVE normally has the **rxSubMap** group variables 1.1 through 2.8 assigned the audio channel values 33 through 48 consecutively. Remember, this is the equivalent of physical audio outputs 1 through 16.

The screenshot shows a window titled "Info 1 - 1 OID groups" with a table of variables. The table has three columns: Name, Syntax, and Value. The variables are listed from rxSubMap.1.1 to rxSubMap.2.8, with values ranging from 33 to 48. The interface also includes a toolbar with icons for refresh, search, and poll, a dropdown menu showing the IP address 192.34.150.38, a poll interval of 60 seconds, and a status bar at the bottom showing 52 objects, the same IP address, SNMPv1 protocol, 161 MIBs, 1 instance, and the last successful poll time.

Name	Syntax	Value
rxSubMap.1.1	int32	33
rxSubMap.1.2	int32	34
rxSubMap.1.3	int32	35
rxSubMap.1.4	int32	36
rxSubMap.1.5	int32	37
rxSubMap.1.6	int32	38
rxSubMap.1.7	int32	39
rxSubMap.1.8	int32	40
rxSubMap.2.1	int32	41
rxSubMap.2.2	int32	42
rxSubMap.2.3	int32	43
rxSubMap.2.4	int32	44
rxSubMap.2.5	int32	45
rxSubMap.2.6	int32	46
rxSubMap.2.7	int32	47
rxSubMap.2.8	int32	48

Figure 4

To configure the RAVE receivers, we open the **receivers** folder at OID 2680.1.1.6. We continue to open the subfolders under **receivers** and then open the **rxSubMap** leaf. In order to route the two 4-channel bundles from the secondary RAVEs to the primary RAVE's eight audio outputs, we must re-assign the audio values accordingly. The first

four channels from device B, unicast bundle 304, are assigned the values 33 through 36. This is the normal and default configuration for the RAVE receivers in group 1. This routes channels 1 through 4, extracted from bundle 304, to the primary RAVE's first four physical outputs. The second four channels, values 41 through 44, are from device C, unicast bundle 305. These values are normally assigned to variables **rxSubMap.2.1** through **rxSubMap.2.4** indicating the first 4 channels of the second bundle. These channels would normally drive physical outputs 9 through 12, such as on a RAVE 160s-24. However, for this application these channels are going to drive the primary RAVE's physical outputs 5 through 8.

To complete the audio configuration, we assign the remaining channels from each bundle to the second group. Unlike transmitters, *all receive channels must be assigned a unique value*. Assigning the unused channels to group 2 discards the garbage that we do not wish to receive. Figure 5 shows the end result of the primary RAVEs receive configuration.

The screenshot shows a window titled 'Info 7' with a toolbar at the top containing a refresh icon, a stop icon, a search icon, an IP address field (191.34.150.42), a poll interval field (60 seconds), and a Log checkbox. Below the toolbar is a table with three columns: Name, Syntax, and Value. The table lists 16 variables, each with a small icon to its left. At the bottom of the window, there is a status bar with several indicators: a green light, the number 54, the IP address 191.34.150.42, a protocol icon (SNMPv1), the numbers 161 and 1, and the text 'Last successful poll at 5/15/01 2:16:46 PM'.

Name	Syntax	Value
rxSubMap.1.1	int32	33
rxSubMap.1.2	int32	34
rxSubMap.1.3	int32	35
rxSubMap.1.4	int32	36
rxSubMap.1.5	int32	41
rxSubMap.1.6	int32	42
rxSubMap.1.7	int32	43
rxSubMap.1.8	int32	44
rxSubMap.2.1	int32	37
rxSubMap.2.2	int32	38
rxSubMap.2.3	int32	39
rxSubMap.2.4	int32	40
rxSubMap.2.5	int32	45
rxSubMap.2.6	int32	46
rxSubMap.2.7	int32	47
rxSubMap.2.8	int32	48

Figure 5

Saving the Configuration

Note that the completion of bundle assignments is assumed in this tutorial. The values assigned to unicast bundles in this application are unimportant and is not related to audio channel configuration. However, proper assignment is required for delivery over

the network. Verify proper bundle configuration of bundle assignments before proceeding.

Once the audio configuration is complete, we can enable “persistence” to permanently store the configuration in the RAVE’s non-volatile memory. We do this by first asserting the **flashPersistEnable** variable with a value of 1. The OID for **flashPersistEnable** is 2680.1.1.2.7. This leaf is located under the **flash** folder, which is the second folder under the **coreManager**.

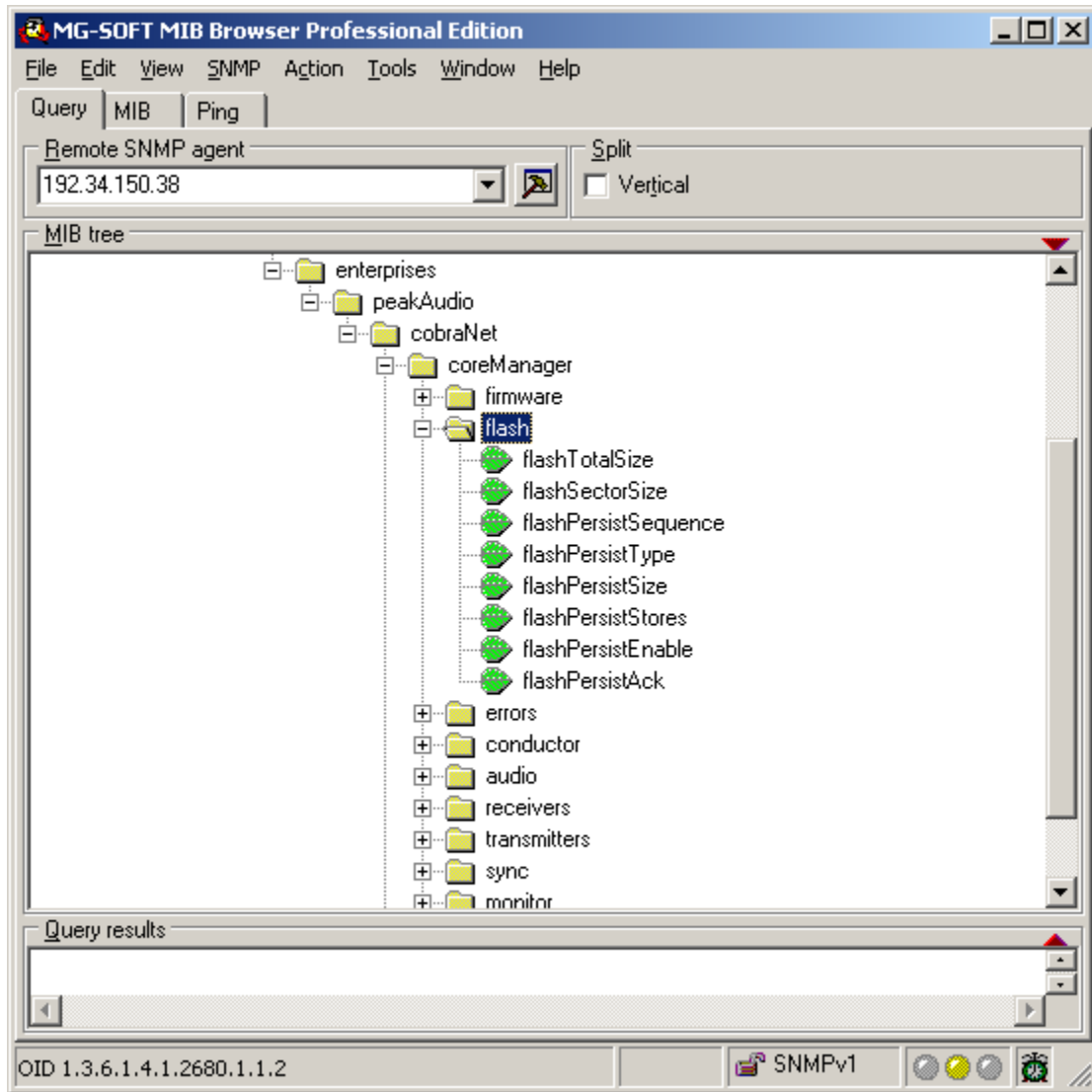


Figure 6

Next, we want to verify that the configuration is written to non-volatile memory so we will force a write to flash. To do this we set the **flashPersistAck** variable to a value of 1. The OID for **flashPersistAck** 2680.1.1.2.8 and is located just below **flashPersistEnable**.

The organization of the peakAudio MIB with the **flash** variables folder opened is shown in Figure 6.

Lockout

After the RAVE is placed into “persistence” mode by enabling the **flashPersistEnable** variable, the unit will no longer respond to configuration through the front panel rotary switches. In “persistence” mode, the RAVE device must be controlled via the software interface.

Lockout can be disabled through hardware by setting all front panel rotary switches to the F position and power cycling the unit. Of course lockout can be disabled through software by simply de-asserting the **flashPersistEnable** variable.