## ZebOS-XP GMRP SMI Reference

IP Infusion Inc.

Generated by Doxygen 1.6.1

Wed Dec 16 12:33:23 2015

# **Contents**

1	Data	a Struct	ture Index		1						
	1.1	Data S	tructures		1						
2	File	Index			3						
	2.1	File Li	ist		3						
3	Data	a Struct	ure Docui	mentation	5						
	3.1	gmrp_	msg_ Stru	ct Reference	5						
	3.2	xmrpB	BridgeConf	fig Struct Reference	6						
	3.3	xmrpC	CommonLi	st Struct Reference	7						
	3.4	xmrpF	smState St	truct Reference	8						
	3.5	xmrpT	xmrpTimers Struct Reference								
	3.6	xmrpV	nrpVlanStats Struct Reference								
4	File	Docum	entation		11						
	4.1	smi_g	mrp.h File	Reference	11						
		4.1.1	Detailed	Description	17						
		4.1.2	Function	Documentation	17						
			4.1.2.1	smi_gmrp_debug_set	17						
			4.1.2.2	smi_gmrp_debug_unset	17						
			4.1.2.3	smi_gmrp_show_debugging	18						
			4.1.2.4	smi_is_gmrp_enabled	18						
			4.1.2.5	smi_mmrp_disable_periodic_timer_global	18						
			4.1.2.6	smi_mmrp_disable_periodic_timer_per_port	19						
			4.1.2.7	smi_mmrp_enable_periodic_timer_global	19						
			4.1.2.8	smi mmrp enable periodic timer per port	20						

ii CONTENTS

		4.1.2.9	smi_mmrp_get_periodic_timer_per_port_status	20
		4.1.2.10	smi_mmrp_set_if_mode_to_p2p	21
		4.1.2.11	smi_mmrp_unset_if_mode_to_p2p	21
		4.1.2.12	smi_mmrp_verify_p2p_mode_set	21
		4.1.2.13	smi_xmrp_clear_all_vlan_statistics	22
		4.1.2.14	smi_xmrp_clear_per_vlan_statistics	22
		4.1.2.15	smi_xmrp_disable_all_port	22
		4.1.2.16	smi_xmrp_disable_extended_filtering	23
		4.1.2.17	smi_xmrp_disable_forward_all	23
		4.1.2.18	smi_xmrp_disable_global	24
		4.1.2.19	smi_xmrp_disable_per_port	24
		4.1.2.20	smi_xmrp_disable_per_port_per_vlan	24
		4.1.2.21	smi_xmrp_disable_per_vlan	25
		4.1.2.22	smi_xmrp_enable_all_port	25
		4.1.2.23	smi_xmrp_enable_extended_filtering	26
		4.1.2.24	smi_xmrp_enable_forward_all	26
		4.1.2.25	smi_xmrp_enable_global	27
		4.1.2.26	smi_xmrp_enable_per_port	27
		4.1.2.27	smi_xmrp_enable_per_port_per_vlan	27
		4.1.2.28	smi_xmrp_enable_per_vlan	28
		4.1.2.29	smi_xmrp_get_extended_filtering_flag	28
		4.1.2.30	smi_xmrp_get_forward_all_flag	29
		4.1.2.31	smi_xmrp_get_registration_type	29
		4.1.2.32	smi_xmrp_get_timer	30
		4.1.2.33	smi_xmrp_set_join_timer	30
		4.1.2.34	smi_xmrp_set_leave_all_timer	31
		4.1.2.35	smi_xmrp_set_leave_timer	31
		4.1.2.36	smi_xmrp_set_registration_type	32
		4.1.2.37	smi_xmrp_show_bridge_configuration	33
		4.1.2.38	smi_xmrp_show_finite_state_machine	33
		4.1.2.39	smi_xmrp_show_per_vlan_statistics	34
		4.1.2.40	smi_xmrp_show_times	34
4.2	smi_gı	mrp_msg.h	File Reference	36
	4.2.1	Detailed	Description	37

# **Chapter 1**

# **Data Structure Index**

### 1.1 Data Structures

Here are the data structures with brief descriptions:

gmrp_msg																		5
xmrpBridgeConfig	3																	6
xmrpCommonList	ţ																	7
xmrpFsmState .																		8
xmrpTimers																		9
xmrpVlanStats .																		

# Chapter 2

# **File Index**

### 2.1 File List

Here is a list of all documented files with brief descriptions:

smi_gmrp.h (Provides APIs for managing GARP Multicast Registration Pro-
• • •
tocol (GMRP) GMRP is a Generic Attribute Registration Protocol
(GARP) application that provides a constrained multicast flooding
facility similar to IGMP snooping. The APIs provided in this file
forms the basis of ZebOS GMRP/MMRP management. These APIs
are used by various north bound management interfaces like CLI,
SNMP and SMI) 1
smi_gmrn_msg h (Defines data structures used by gmrn/mmrn SMI APIs )

4 File Index

## **Chapter 3**

## **Data Structure Documentation**

### 3.1 gmrp\_msg\_ Struct Reference

#### **Data Fields**

- smi\_cindex\_t cindex\_0
- u\_int32\_t **vr\_id**
- char protocol [255]
- char bridge\_name [255]
- u\_int16\_t vlan\_id
- char **ifname** [255]
- pal\_time\_t timer\_value
- char reg\_type [255]
- u\_char is\_gmrp\_en
- int timer\_type
- u\_char is\_p2p
- u\_char periodic\_timer\_status
- int regis\_type
- int forward\_all
- int extended\_filtering\_flag
- int smi\_debug
- struct xmrpTimers timerData
- struct xmrpVlanStats statsData
- struct xmrpCommonList xmrpData

The documentation for this struct was generated from the following file:

## 3.2 xmrpBridgeConfig Struct Reference

#### **Data Fields**

- char portName [32]
- char ifState
- u\_char registrationType
- u\_char forwardAll
- pal\_time\_t timerValues [SMI\_GMRP\_GARP\_MAX\_TIMERS]

The documentation for this struct was generated from the following file:

## 3.3 xmrpCommonList Struct Reference

#### **Data Fields**

- int have\_more
- int more\_count
- struct list \* xmrpList

The documentation for this struct was generated from the following file:

### 3.4 xmrpFsmState Struct Reference

#### **Data Fields**

- char **portName** [32]
- u\_int16\_t vlanId
- int numState
- char attrIndex [128]
- char applicantState [128][4]
- char registrarState [128][4]

The documentation for this struct was generated from the following file:

## 3.5 xmrpTimers Struct Reference

#### **Data Fields**

• pal\_time\_t timerValues [SMI\_GMRP\_GARP\_MAX\_TIMERS]

The documentation for this struct was generated from the following file:

## 3.6 xmrpVlanStats Struct Reference

#### **Data Fields**

- u\_int32\_t rxCounters [XMRP\_TOTAL\_ATTR\_EVENTS]
- u\_int32\_t txCounters [XMRP\_TOTAL\_ATTR\_EVENTS]

The documentation for this struct was generated from the following file:

## **Chapter 4**

## **File Documentation**

### 4.1 smi\_gmrp.h File Reference

Provides APIs for managing GARP Multicast Registration Protocol (GMRP) GMRP is a Generic Attribute Registration Protocol (GARP) application that provides a constrained multicast flooding facility similar to IGMP snooping. The APIs provided in this file forms the basis of ZebOS GMRP/MMRP management. These APIs are used by various north bound management interfaces like CLI, SNMP and SMI. #include "smi\_client.h"

```
#include "smi_gmrp_msg.h"
```

#### **Functions**

- int smi\_client\_create\_n\_send\_gmrp\_msg (struct smi\_client\_handler \*async, int vrid, gmrp\_msg \*msg, int optype)
- int **smi\_client\_read\_sync\_gmrp\_msg** (struct smi\_client\_handler \*ach, int ms-gtype, void \*getmsg)
- s\_int32\_t smi\_xmrp\_enable\_global (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*bridge\_name)

Enable the GMRP/MMRP feature globally for the given bridge instance. NOTE: GMRP cannot be enabled if IGMP snooping is enabled.

- s\_int32\_t smi\_xmrp\_enable\_global\_validate (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*bridge\_name)
- s\_int32\_t smi\_xmrp\_disable\_global (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*bridge\_name)

Disable the GMRP/MMRP feature globally for the given bridge instance.

- s\_int32\_t smi\_xmrp\_disable\_global\_validate (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*bridge\_name)
- s\_int32\_t smi\_xmrp\_enable\_per\_vlan (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*bridge\_name, u\_int16\_t vlan\_id)

Enable the GMRP/MMRP feature per VLAN for the given bridge instance. NOTE: GMRP cannot be enabled if IGMP snooping is enabled.

- s\_int32\_t smi\_xmrp\_enable\_per\_vlan\_validate (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*bridge\_name, u\_int16\_t vlan\_id)
- s\_int32\_t smi\_xmrp\_disable\_per\_vlan (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*bridge\_name, u\_int16\_t vlan\_id)

Disable the GMRP/MMRP feature per VLAN for the given bridge instance. NOTE: GMRP cannot be enabled if IGMP snooping is enabled.

- s\_int32\_t smi\_xmrp\_disable\_per\_vlan\_validate (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*bridge\_name, u\_int16\_t vlan\_id)
- s\_int32\_t smi\_xmrp\_enable\_per\_port\_per\_vlan (struct smiclient\_globals \*azg, u int32 t vr id, char \*protocol, char \*ifname, u int16 t vlan id)

Enables the GMRP/MMRP for the particular port instance of particular VLAN and starts required timers. NOTE: GMRP cannot be enabled if IGMP snooping is enabled.

- s\_int32\_t smi\_xmrp\_enable\_per\_port\_per\_vlan\_validate (struct smiclient\_-globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*ifname, u\_int16\_t vlan\_id)
- s\_int32\_t smi\_xmrp\_disable\_per\_port\_per\_vlan (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*ifname, u\_int16\_t vlan\_id)

Disables the GMRP/MMRP for the particular port instance of particular VLAN and starts required timers. NOTE: GMRP cannot be enabled if IGMP snooping is enabled.

- s\_int32\_t smi\_xmrp\_disable\_per\_port\_per\_vlan\_validate (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*ifname, u\_int16\_t vlan\_id)
- s\_int32\_t smi\_xmrp\_set\_join\_timer (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*ifname, pal\_time\_t timer\_value)

Sets the join timer, used for joining the group, measured in centiseconds used in GM-RP/MMRP protocol at interface level.

The following types of timers are supported

Join: Specify the timer for joining the group

Leave: Specify the timer for leaving the group

Join: LeaveAll Specify the timer for leaving all groups

- s\_int32\_t smi\_xmrp\_set\_join\_timer\_validate (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*ifname, pal\_time\_t timer\_value)
- s\_int32\_t smi\_xmrp\_set\_leave\_timer (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*ifname, pal\_time\_t timer\_value)

Sets the leave timer, used for leaving the group, measured in centiseconds used in GMRP/MMRP protocol at interface level.

The following types of timers are supported

Join: Specify the timer for joining the group

Leave: Specify the timer for leaving the group Join: LeaveAll Specify the timer for leaving all groups

• s\_int32\_t smi\_xmrp\_set\_leave\_timer\_validate (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*ifname, pal\_time\_t timer\_value)

• s\_int32\_t smi\_xmrp\_set\_leave\_all\_timer (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*ifname, pal\_time\_t timer\_value)

Sets the leave timer, used for leaving from all groups, measured in centiseconds used in GMRP/MMRP protocol at interface level.

The following types of timers are supported

Join: Specify the timer for joining the group

Leave: Specify the timer for leaving the group

Join: LeaveAll Specify the timer for leaving all groups

• s\_int32\_t smi\_xmrp\_set\_leave\_all\_timer\_validate (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*ifname, pal\_time\_t timer\_value)

• s\_int32\_t smi\_mmrp\_set\_if\_mode\_to\_p2p (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*ifname)

Set the interface mode to point-to-point behavior option on an MMRP-enabled bridge.

- s\_int32\_t smi\_mmrp\_set\_if\_mode\_to\_p2p\_validate (struct smiclient\_globals \*azg, u int32 t vr id, char \*ifname)
- s\_int32\_t smi\_mmrp\_unset\_if\_mode\_to\_p2p (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*ifname)

Unset the interface mode to point-to-point behavior option on an MMRP-enabled bridge.

- s\_int32\_t **smi\_mmrp\_unset\_if\_mode\_to\_p2p\_validate** (struct smiclient\_-globals \*azg, u\_int32\_t vr\_id, char \*ifname)
- s\_int32\_t smi\_mmrp\_enable\_periodic\_timer\_global (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*bridge\_name)

Enable the periodic timer option for all ports of an MMRP-enabled bridge.

- s\_int32\_t smi\_mmrp\_disable\_periodic\_timer\_global (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*bridge\_name)

Enable the periodic timer option for all ports of an MMRP-enabled bridge.

- s\_int32\_t **smi\_mmrp\_disable\_periodic\_timer\_global\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*bridge\_name)
- s\_int32\_t smi\_mmrp\_enable\_periodic\_timer\_per\_port (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*ifname)

Enable the periodic timer option for the given port of an MMRP-enabled bridge.

• s\_int32\_t **smi\_mmrp\_enable\_periodic\_timer\_per\_port\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*ifname)

• s\_int32\_t smi\_mmrp\_disable\_periodic\_timer\_per\_port (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*ifname)

Disable the periodic timer option for the given port of an MMRP-enabled bridge.

- s\_int32\_t **smi\_mmrp\_disable\_periodic\_timer\_per\_port\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*ifname)
- s\_int32\_t smi\_xmrp\_enable\_per\_port (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*ifname)

Enables the GMRP/MMRP on the particular port instance of all VLANs and starts required timers.

- s\_int32\_t smi\_xmrp\_enable\_per\_port\_validate (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*ifname)
- s\_int32\_t smi\_xmrp\_enable\_all\_port (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol)

Enables the GMRP/MMRP on all the ports instance of all VLANs and starts required timers.

- s\_int32\_t smi\_xmrp\_enable\_all\_port\_validate (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol)
- s\_int32\_t smi\_xmrp\_disable\_per\_port (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*ifname)

Disables the GMRP/MMRP on the particular port instance of all VLANs and starts required timers.

- s\_int32\_t **smi\_xmrp\_disable\_per\_port\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*ifname)
- s\_int32\_t smi\_xmrp\_disable\_all\_port (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol)

Disables the GMRP/MMRP on all the ports instance of all VLANs and starts required timers.

- s\_int32\_t smi\_xmrp\_disable\_all\_port\_validate (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol)
- s\_int32\_t smi\_xmrp\_set\_registration\_type (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*ifname, char \*reg\_type)

Sets the GMRP/MMRP port's multicast group registration type to one of the following. fixed: The multicast groups currently registered on the switch will remain on the port, but subsequent new registrations or de-registrations based on timers do not affect the port.

forbidden: All registered multicast groups are de-registered, and prevents further multicast registration on the port.

normal: Set multicast group registration and de-registration to dynamic.

restricted: Set to restricted registration.

NOTE: GMRP cannot be enabled if IGMP snooping is enabled.

- s\_int32\_t smi\_xmrp\_set\_registration\_type\_validate (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*ifname, char \*reg\_type)
- s\_int32\_t smi\_xmrp\_enable\_forward\_all (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*ifname)

Enable GMRP/MMRP protocol's 'forwrd all' option on the interface.

- s\_int32\_t smi\_xmrp\_enable\_forward\_all\_validate (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*ifname)
- s\_int32\_t smi\_xmrp\_disable\_forward\_all (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*ifname)

Disable GMRP/MMRP protocol's 'forwrd all' option on the interface.

- s\_int32\_t smi\_xmrp\_disable\_forward\_all\_validate (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*ifname)
- s\_int32\_t smi\_xmrp\_enable\_extended\_filtering (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*bridge\_name)

Enable GMRP/MMRP to enable 'extended filtering option at bridge level.

- s\_int32\_t **smi\_xmrp\_enable\_extended\_filtering\_validate** (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*bridge\_name)
- s\_int32\_t smi\_xmrp\_disable\_extended\_filtering (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*bridge\_name)

Disable GMRP/MMRP to enable 'extended filtering option at bridge level.

- s\_int32\_t smi\_xmrp\_disable\_extended\_filtering\_validate (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*bridge\_name)
- s\_int32\_t smi\_xmrp\_clear\_all\_vlan\_statistics (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*bridge\_name)

Clear GMRP/MMRP statistics of all VLANs.

- s\_int32\_t smi\_xmrp\_clear\_all\_vlan\_statistics\_validate (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*bridge\_name)
- s\_int32\_t smi\_xmrp\_clear\_per\_vlan\_statistics (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, u\_int16\_t vlan\_id, char \*bridge\_name)

Clear GMRP/MMRP statistics of all VLANs.

• s\_int32\_t smi\_xmrp\_show\_times (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*ifname, struct xmrpTimers \*timerData)

Show GMRP/MMRP configured timer values.

• s\_int32\_t smi\_xmrp\_show\_bridge\_configuration (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*bridge\_name, struct list \*brConfList, int(\*funPointer)(struct list \*brConfList))

Show GMRP/MMRP bridge configuration.

• s\_int32\_t smi\_xmrp\_show\_finite\_state\_machine (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*bridge\_name, struct list \*fsmStateList, int(\*funPointer)(struct list \*fsmStateList))

Show GMRP/MMRP finite state machine details.

• s\_int32\_t smi\_xmrp\_show\_per\_vlan\_statistics (struct smiclient\_globals \*azg, u\_int32\_t vr\_id, char \*protocol, char \*bridge\_name, u\_int16\_t vlan\_id, struct xmrpVlanStats \*vlanStats)

Show various GMRP/MMRP related statistics of particular VLAN.

• int smi\_is\_gmrp\_enabled (struct smiclient\_globals \*azg, int vr\_id, char \*protocol, char \*bridge\_name, u\_char \*is\_gmrp\_en)

Function to verify whether GMRP is enabled globally.

• int smi\_xmrp\_get\_timer (struct smiclient\_globals \*azg, int vr\_id, char \*protocol, char \*ifname, int timer\_type, pal\_time\_t \*timer\_value)

Function to get configured timer value for the specific timer.

• int smi\_mmrp\_verify\_p2p\_mode\_set (struct smiclient\_globals \*azg, int vr\_id, char \*ifname, u\_char \*is\_p2p)

Function to verify whether p2p mode is enable on port.

- int smi\_mmrp\_get\_periodic\_timer\_per\_port\_status (struct smiclient\_globals \*azg, int vr\_id, char \*ifname, u\_char \*periodic\_timer\_status)
  - Function to verify periodic timer is set for the specific port.
- int smi\_xmrp\_get\_registration\_type (struct smiclient\_globals \*azg, int vr\_id, char \*protocol, char \*ifname, int \*regis\_type)

Function to get muticast group registration type.

• int smi\_xmrp\_get\_forward\_all\_flag (struct smiclient\_globals \*azg, int vr\_id, char \*protocol, char \*ifname, int \*forward\_all)

 $Function \ to \ verify \ fow ard \ all \ flag \ is \ enabled.$ 

- int smi\_xmrp\_get\_extended\_filtering\_flag (struct smiclient\_globals \*azg, int vr\_id, char \*protocol, char \*bridge\_name, int \*extended\_filtering\_flag)

  Function to get extended filtering flag.
- int smi\_gmrp\_debug\_set (struct smiclient\_globals \*azg, int vr\_id, int debug)
   sets for debug value
- int smi\_gmrp\_debug\_unset (struct smiclient\_globals \*azg, int vr\_id, int debug)
   unset for debug value
- int smi\_gmrp\_show\_debugging (struct smiclient\_globals \*azg, int vr\_id, u\_int32\_t \*smi\_debug)
   show deebug events
- int **smi\_gmrp\_debug\_set\_validate** (struct smiclient\_globals \*azg, u\_int32\_- t debug)

int smi\_gmrp\_debug\_unset\_validate (struct smiclient\_globals \*azg, u\_int32\_t debug)

#### 4.1.1 Detailed Description

Provides APIs for managing GARP Multicast Registration Protocol (GMRP) GMRP is a Generic Attribute Registration Protocol (GARP) application that provides a constrained multicast flooding facility similar to IGMP snooping. The APIs provided in this file forms the basis of ZebOS GMRP/MMRP management. These APIs are used by various north bound management interfaces like CLI, SNMP and SMI.

#### **4.1.2** Function Documentation

## **4.1.2.1** int smi\_gmrp\_debug\_set (struct smiclient\_globals \* azg, int vr\_id, int debug)

sets for debug value smi\_gmrp\_debug\_set

#### **Parameters:**

- $\leftarrow$  azg Pointer to the SMI client global structure
- $\leftarrow vr\_id$
- ← debug

#### **Returns:**

0 on success RESULT\_OK on success

# 4.1.2.2 int smi\_gmrp\_debug\_unset (struct smiclient\_globals \* azg, int $vr\_id$ , int debug)

unset for debug value smi\_gmrp\_debug\_unset

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- $\leftarrow vr\_id$
- $\leftarrow$  debug

#### **Returns:**

0 on success RESULT\_OK on success

## 4.1.2.3 int smi\_gmrp\_show\_debugging (struct smiclient\_globals \* azg, int vr\_id, u\_int32\_t \* smi\_debug)

show deebug events smi\_gmrp\_show\_debugging

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- $\leftarrow vr id$
- → smi\_debug

#### **Returns:**

0 on success RESULT\_OK on success

## 4.1.2.4 int smi\_is\_gmrp\_enabled (struct smiclient\_globals \* azg, int vr\_id, char \* protocol, char \* bridge\_name, u\_char \* is\_gmrp\_en)

Function to verify whether GMRP is enabled globally. smi\_is\_gmrp\_enabled

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- ← vr\_id Virtual Router ID <0-255>
- ← *protocol* Protocol type string {gmrp|mmrp}
- ← *bridge\_name* Bridge name <1-32>
- → is\_gmrp\_en Flag to mention whether gmrp enabled or disabled Enabled (1)
   Disable (0)

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_GENERAL

## 4.1.2.5 s\_int32\_t smi\_mmrp\_disable\_periodic\_timer\_global (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* bridge\_name)

Enable the periodic timer option for all ports of an MMRP-enabled bridge.  $smi_mrp_disable_periodic_timer_global$ 

- ← azg Pointer to the SMI client global structure
- $\leftarrow vr \ id \ Virtual Router ID < 0-255 >$
- ← *bridge\_name* Bridge name

#### **Returns:**

```
0 on success, otherwise one of the following error codes
```

NSM\_BRIDGE\_ERR\_NOTFOUND

NSM\_BRIDGE\_ERR\_SPBD\_TYPE

NSM BRIDGE NO PORT CFG

NSM\_BRIDGE\_ERR\_NOT\_BOUND

NSM\_BRIDGE\_ERR\_MMRP\_NOCONFIG

NSM\_ERR\_MMRP\_NOCONFIG\_ONPORT

## 4.1.2.6 s\_int32\_t smi\_mmrp\_disable\_periodic\_timer\_per\_port (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* ifname)

Disable the periodic timer option for the given port of an MMRP-enabled bridge. smi\_mmrp\_disable\_periodic\_timer\_per\_port

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- ← *vr\_id* Virtual Router ID <0-255>
- *← ifname* Interface name

#### **Returns:**

0 on success, otherwise one of the following error codes

NSM\_BRIDGE\_ERR\_NOTFOUND

NSM\_BRIDGE\_ERR\_SPBD\_TYPE

NSM\_BRIDGE\_NO\_PORT\_CFG

NSM\_BRIDGE\_ERR\_NOT\_BOUND

NSM\_BRIDGE\_ERR\_MMRP\_NOCONFIG NSM\_ERR\_MMRP\_NOCONFIG\_ONPORT

.1.2.7 s\_int32\_t smi\_mmrp\_enable\_periodic\_timer\_global (struct

## smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* bridge\_name)

Enable the periodic timer option for all ports of an MMRP-enabled bridge. smi\_mmrp\_enable\_periodic\_timer\_global

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- ← *vr\_id* Virtual Router ID <0-255>
- ← bridge\_name Bridge name

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_NOTFOUND

```
NSM_BRIDGE_ERR_SPBD_TYPE
NSM_BRIDGE_NO_PORT_CFG
NSM_BRIDGE_ERR_NOT_BOUND
NSM_BRIDGE_ERR_MMRP_NOCONFIG
NSM_ERR_MMRP_NOCONFIG_ONPORT
```

## 4.1.2.8 s\_int32\_t smi\_mmrp\_enable\_periodic\_timer\_per\_port (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* ifname)

Enable the periodic timer option for the given port of an MMRP-enabled bridge. smi\_mmrp\_enable\_periodic\_timer\_per\_port

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- ← *vr\_id* Virtual Router ID <0-255>
- ← *ifname* Interface name

#### **Returns:**

0 on success, otherwise one of the following error codes

NSM\_BRIDGE\_ERR\_NOTFOUND

NSM\_BRIDGE\_ERR\_SPBD\_TYPE

NSM\_BRIDGE\_NO\_PORT\_CFG

NSM BRIDGE ERR NOT BOUND

NSM\_BRIDGE\_ERR\_MMRP\_NOCONFIG

NSM\_ERR\_MMRP\_NOCONFIG\_ONPORT

# 4.1.2.9 int smi\_mmrp\_get\_periodic\_timer\_per\_port\_status (struct smiclient\_globals \* azg, int vr\_id, char \* ifname, u\_char \* periodic\_timer\_status)

Function to verify periodic timer is set for the specific port. smi\_mmrp\_get\_periodic\_timer\_per\_port\_status

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- $\leftarrow vr \ id \ \text{Virtual Router ID} < 0-255 >$
- ← *ifname* Interface Name
- → *periodic\_timer\_status* timer status

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_GENERAL

## 4.1.2.10 s\_int32\_t smi\_mmrp\_set\_if\_mode\_to\_p2p (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* ifname)

Set the interface mode to point-to-point behavior option on an MMRP-enabled bridge.  $smi\_mmrp\_set\_if\_mode\_to\_p2p$ 

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- ← *vr\_id* Virtual Router ID <0-255>
- ← *ifname* Interface name

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_GENERAL

## 4.1.2.11 s\_int32\_t smi\_mmrp\_unset\_if\_mode\_to\_p2p (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* ifname)

Unset the interface mode to point-to-point behavior option on an MMRP-enabled bridge. smi\_mmrp\_unset\_if\_mode\_to\_p2p

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- $\leftarrow vr\_id$  Virtual Router ID <0-255>
- *← ifname* Interface name

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_GENERAL

## 4.1.2.12 int smi\_mmrp\_verify\_p2p\_mode\_set (struct smiclient\_globals \* azg, int vr\_id, char \* ifname, u\_char \* is\_p2p)

Function to verify whether p2p mode is enable on port. smi\_mmrp\_verify\_p2p\_mode\_set

#### Parameters:

- ← azg Pointer to the SMI client global structure
- $\leftarrow vr \ id \ Virtual Router ID < 0-255 >$
- $\leftarrow$  *ifname* Interface Name
- $\rightarrow$  *is\_p2p* 1 mode p2p

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_GENERAL

## 4.1.2.13 s\_int32\_t smi\_xmrp\_clear\_all\_vlan\_statistics (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol, char \* bridge\_name)

Clear GMRP/MMRP statistics of all VLANs. smi\_xmrp\_clear\_all\_vlan\_statistics

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- $\leftarrow vr \ id \ \text{Virtual Router ID} < 0-255 >$
- ← *protocol* Protocol type string {gmrp|mmrp}
- ← *bridge name* Bridge name <1-32>

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_GENERAL

# 4.1.2.14 s\_int32\_t smi\_xmrp\_clear\_per\_vlan\_statistics (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol, u\_int16\_t vlan\_id, char \* bridge\_name)

Clear GMRP/MMRP statistics of all VLANs. smi\_xmrp\_clear\_per\_vlan\_statistics

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- ← *vr\_id* Virtual Router ID <0-255>
- ← *protocol* Protocol type string {gmrp|mmrp}
- ← *vlan\_id* VLAN number <1-4094>
- ← *bridge\_name* Bridge name <1-32>

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_GENERAL

## 4.1.2.15 s\_int32\_t smi\_xmrp\_disable\_all\_port (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol)

Disables the GMRP/MMRP on all the ports instance of all VLANs and starts required timers. smi\_xmrp\_disable\_all\_port

- ← azg Pointer to the SMI client global structure
- ← vr\_id Virtual Router ID <0-255>

← *protocol* Protocol type string {gmrp|mmrp}

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_NOT\_CFG NSM\_BRIDGE\_NO\_PORT\_CFG

# 4.1.2.16 s\_int32\_t smi\_xmrp\_disable\_extended\_filtering (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol, char \* bridge\_name)

Disable GMRP/MMRP to enable 'extended filtering option at bridge level. smi\_xmrp\_disable\_extended\_filtering

#### Parameters:

- $\leftarrow$  azg Pointer to the SMI client global structure
- ← *vr\_id* Virtual Router ID <0-255>
- ← *protocol* Protocol type string {gmrp|mmrp}
- ← *bridge\_name* Bridge name <1-32>

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_NOTFOUND NSM\_VLAN\_ERR\_BRIDGE\_NOT\_VLAN\_AWARE

## 4.1.2.17 s\_int32\_t smi\_xmrp\_disable\_forward\_all (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol, char \* ifname)

Disable GMRP/MMRP protocol's 'forwrd all' option on the interface. smi\_xmrp\_disable\_forward\_all

#### **Parameters:**

- $\leftarrow$  azg Pointer to the SMI client global structure
- ← *vr\_id* Virtual Router ID <0-255>
- ← *protocol* Protocol type string {gmrp|mmrp}
- ← *ifname* Interface name

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_ERR\_GMRP\_NOCONFIG\_ONPORT NSM\_BRIDGE\_ERR\_GENERAL

## 4.1.2.18 s\_int32\_t smi\_xmrp\_disable\_global (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol, char \* bridge\_name)

Disable the GMRP/MMRP feature globally for the given bridge instance. smi\_xmrp\_disable\_global

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- $\leftarrow vr \ id \ Virtual Router ID < 0-255 >$
- ← *protocol* Protocol type string {gmrp|mmrp}
- ← *bridge\_name* Bridge name <1-32>

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_NOTFOUND

## 4.1.2.19 s\_int32\_t smi\_xmrp\_disable\_per\_port (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol, char \* ifname)

Disables the GMRP/MMRP on the particular port instance of all VLANs and starts required timers. smi\_xmrp\_disable\_per\_port

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- ← *vr\_id* Virtual Router ID <0-255>
- ← *protocol* Protocol type string {gmrp|mmrp}
- ← *ifname* Interface name

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_GENERAL NSM\_BRIDGE\_ERR\_SPBD\_TYPE

# 4.1.2.20 s\_int32\_t smi\_xmrp\_disable\_per\_port\_per\_vlan (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol, char \* ifname, u\_int16\_t vlan\_id)

Disables the GMRP/MMRP for the particular port instance of particular VLAN and starts required timers. NOTE: GMRP cannot be enabled if IGMP snooping is enabled. smi\_xmrp\_disable\_per\_port\_per\_vlan

#### **Parameters:**

← azg Pointer to the SMI client global structure

- $\leftarrow vr \ id \ Virtual Router ID < 0-255 >$
- ← *protocol* Protocol type string {gmrp|mmrp}
- ← *ifname* Interface name
- ← *vlan\_id* VLAN number <1-4094>

#### **Returns:**

0 on success, otherwise one of the following error codes

NSM\_BRIDGE\_ERR\_GENERAL

NSM BRIDGE ERR GMRP NOCONFIG

NSM\_GMRP\_ERR\_GMRP\_NOT\_CFG\_ON\_VLAN

NSM\_GMRP\_ERR\_VLAN\_NOT\_CFG\_ON\_PORT

NSM ERR GMRP NOCONFIG ONPORT

NSM GMRP ERR GMRP NOT CFG ON PORT VLAN

NSM\_GMRP\_ERR\_GMRP\_GLOBAL\_CFG\_PORT

# 4.1.2.21 s\_int32\_t smi\_xmrp\_disable\_per\_vlan (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol, char \* bridge\_name, u\_int16\_t vlan id)

Disable the GMRP/MMRP feature per VLAN for the given bridge instance. NOTE: GMRP cannot be enabled if IGMP snooping is enabled. smi\_xmrp\_disable\_per\_vlan

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- $\leftarrow vr \ id \ \text{Virtual Router ID} < 0-255 >$
- ← *protocol* Protocol type string {gmrp|mmrp}
- ← *bridge\_name* Bridge name <1-32>
- $\leftarrow$  *vlan id* VLAN number <1-4094>

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_NOTFOUND NSM\_BRIDGE\_ERR\_MEM

## 4.1.2.22 s\_int32\_t smi\_xmrp\_enable\_all\_port (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol)

Enables the GMRP/MMRP on all the ports instance of all VLANs and starts required timers. smi\_xmrp\_enable\_all\_port

#### Parameters:

 $\leftarrow$  azg Pointer to the SMI client global structure

```
    ← vr_id Virtual Router ID <0-255>
    ← protocol Protocol type string {gmrp|mmrp}
```

#### **Returns:**

```
0 on success, otherwise one of the following error codes NSM_BRIDGE_NOT_CFG NSM_BRIDGE_NO_PORT_CFG
```

# 4.1.2.23 s\_int32\_t smi\_xmrp\_enable\_extended\_filtering (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol, char \* bridge\_name)

Enable GMRP/MMRP to enable 'extended filtering option at bridge level. smi\_xmrp\_enable\_extended\_filtering

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- $\leftarrow vr \ id \ Virtual Router ID < 0-255 >$
- ← *protocol* Protocol type string {gmrp|mmrp}
- ← *bridge name* Bridge name <1-32>

#### **Returns:**

```
0 on success, otherwise one of the following error codes NSM_BRIDGE_ERR_NOTFOUND NSM_VLAN_ERR_BRIDGE_NOT_VLAN_AWARE
```

## 4.1.2.24 s\_int32\_t smi\_xmrp\_enable\_forward\_all (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol, char \* ifname)

Enable GMRP/MMRP protocol's 'forwrd all' option on the interface. smi\_xmrp\_enable\_forward\_all

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- ← *vr\_id* Virtual Router ID <0-255>
- ← *protocol* Protocol type string {gmrp|mmrp}
- *← ifname* Interface name

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_ERR\_GMRP\_NOCONFIG\_ONPORT NSM\_BRIDGE\_ERR\_GENERAL

## 4.1.2.25 s\_int32\_t smi\_xmrp\_enable\_global (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol, char \* bridge\_name)

Enable the GMRP/MMRP feature globally for the given bridge instance. NOTE: GMRP cannot be enabled if IGMP snooping is enabled. smi\_xmrp\_enable\_global

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- ← *vr\_id* Virtual Router ID <0-255>
- $\leftarrow$  *protocol* Protocol type string {gmrp|mmrp}
- ← *bridge\_name* Bridge name <1-32>

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_NOTFOUND NSM\_BRIDGE\_ERR\_MEM

## 4.1.2.26 s\_int32\_t smi\_xmrp\_enable\_per\_port (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol, char \* ifname)

Enables the GMRP/MMRP on the particular port instance of all VLANs and starts required timers. smi\_xmrp\_enable\_per\_port

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- ← vr\_id Virtual Router ID <0-255>
- ← *protocol* Protocol type string {gmrp|mmrp}
- *← ifname* Interface name

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_GENERAL NSM\_BRIDGE\_ERR\_SPBD\_TYPE

# 4.1.2.27 s\_int32\_t smi\_xmrp\_enable\_per\_port\_per\_vlan (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol, char \* ifname, u\_int16\_t vlan\_id)

Enables the GMRP/MMRP for the particular port instance of particular VLAN and starts required timers. NOTE: GMRP cannot be enabled if IGMP snooping is enabled. smi\_xmrp\_enable\_per\_port\_per\_vlan

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- $\leftarrow vr \ id \ \text{Virtual Router ID} < 0-255 >$
- $\leftarrow$  *protocol* Protocol type string {gmrp|mmrp}
- ← *ifname* Interface name
- ← *vlan\_id* VLAN number <1-4094>

#### **Returns:**

0 on success, otherwise one of the following error codes

NSM\_BRIDGE\_ERR\_GENERAL

NSM\_BRIDGE\_ERR\_GMRP\_NOCONFIG

NSM\_GMRP\_ERR\_GMRP\_NOT\_CFG\_ON\_VLAN

NSM\_GMRP\_ERR\_VLAN\_NOT\_CFG\_ON\_PORT

NSM\_GMRP\_ERR\_GMRP\_GLOBAL\_CFG\_PORT

4.1.2.28 s\_int32\_t smi\_xmrp\_enable\_per\_vlan (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol, char \* bridge\_name, u\_int16\_t vlan\_id)

Enable the GMRP/MMRP feature per VLAN for the given bridge instance. NOTE: GMRP cannot be enabled if IGMP snooping is enabled. smi\_xmrp\_enable\_per\_vlan

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- ← vr\_id Virtual Router ID <0-255>
- ← *protocol* Protocol type string {gmrp|mmrp}
- ← *bridge\_name* Bridge name <1-32>
- $\leftarrow$  *vlan\_id* VLAN number <1-4094>

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_NOTFOUND NSM\_BRIDGE\_ERR\_MEM

4.1.2.29 int smi\_xmrp\_get\_extended\_filtering\_flag (struct smiclient\_globals \* azg, int vr\_id, char \* protocol, char \* bridge\_name, int \* extended\_filtering\_flag)

Function to get extended filtering flag. smi\_xmrp\_get\_extended\_filtering\_flag

#### **Parameters:**

← azg Pointer to the SMI client global structure

```
    ← vr_id Virtual Router ID <0-255>
    ← protocol protocol
    ← ifname Interface Name
    → extended_filtering_flag One of the following values enabled (1) disabled (0)
```

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_GENERAL

## 4.1.2.30 int smi\_xmrp\_get\_forward\_all\_flag (struct smiclient\_globals \* azg, int vr\_id, char \* protocol, char \* ifname, int \* forward\_all)

Function to verify foward all flag is enabled. smi\_xmrp\_get\_forward\_all\_flag

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- $\leftarrow vr\_id$  Virtual Router ID <0-255>
- $\leftarrow protocol$  protocol
- ← *ifname* Interface Name
- → *forward\_all* One of the following values

Forward all enabled (1)

Forward all disabled (0)

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_GENERAL

## 4.1.2.31 int smi\_xmrp\_get\_registration\_type (struct smiclient\_globals \* azg, int vr\_id, char \* protocol, char \* ifname, int \* regis\_type)

Function to get muticast group registration type. smi\_xmrp\_get\_registration\_type

- $\leftarrow$  azg Pointer to the SMI client global structure
- ← *vr\_id* Virtual Router ID <0-255>
- $\leftarrow protocol$  protocol
- ← *ifname* Interface Name

```
→ regis_type One of the following values

GID_EVENT_NORMAL_REGISTRATION (8)

GID_EVENT_FIXED_REGISTRATION (9)

GID_EVENT_FORBID_REGISTRATION (10)

GID_EVENT_RESTRICTED_GROUP_REGISTRATION (37)
```

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_GENERAL

## 4.1.2.32 int smi\_xmrp\_get\_timer (struct smiclient\_globals \* azg, int vr\_id, char \* protocol, char \* ifname, int timer\_type, pal\_time\_t \* timer\_value)

Function to get configured timer value for the specific timer. smi\_xmrp\_get\_timer

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- ← vr\_id Virtual Router ID <0-255>
- ← *protocol* Protocol type string {gmrp|mmrp}
- ← *ifname* Interface Name
- timer\_type Pass one of the following timer type SMI\_GARP\_JOIN\_TIMER SMI\_GARP\_LEAVE\_TIMER SMI\_GARP\_LEAVE\_ALL\_TIMER
- → *timer\_value* timer value

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_GENERAL

# 4.1.2.33 s\_int32\_t smi\_xmrp\_set\_join\_timer (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol, char \* ifname, pal\_time\_t timer\_value)

Sets the join timer, used for joining the group, measured in centiseconds used in GM-RP/MMRP protocol at interface level.

The following types of timers are supported

Join: Specify the timer for joining the group

Leave: Specify the timer for leaving the group

Join: LeaveAll Specify the timer for leaving all groups

. smi\_xmrp\_set\_join\_timer

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- ← *vr\_id* Virtual Router ID <0-255>
- ← *protocol* Protocol type string {gmrp|mmrp}
- *← ifname* Interface name
- ← *timer\_value* Timer value in centiseconds <1-0xffffffff>

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_ERR\_GMRP\_NOCONFIG\_ONPORT NSM\_BRIDGE\_ERR\_GENERAL

4.1.2.34 s\_int32\_t smi\_xmrp\_set\_leave\_all\_timer (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol, char \* ifname, pal\_time\_t timer\_value)

Sets the leave timer, used for leaving from all groups, measured in centiseconds used in GMRP/MMRP protocol at interface level.

The following types of timers are supported

Join: Specify the timer for joining the group

Leave: Specify the timer for leaving the group

Join: LeaveAll Specify the timer for leaving all groups

. smi\_xmrp\_set\_leave\_all\_timer

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- $\leftarrow vr \ id \ \text{Virtual Router ID} < 0-255 >$
- ← *protocol* Protocol type string {gmrp|mmrp}
- *← ifname* Interface name
- ← *timer\_value* Timer value in centiseconds <1-0xffffffff>

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_ERR\_GMRP\_NOCONFIG\_ONPORT NSM\_BRIDGE\_ERR\_GENERAL

4.1.2.35 s\_int32\_t smi\_xmrp\_set\_leave\_timer (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol, char \* ifname, pal\_time\_t timer\_value)

Sets the leave timer, used for leaving the group, measured in centiseconds used in GMRP/MMRP protocol at interface level.

The following types of timers are supported

Join: Specify the timer for joining the group

Leave: Specify the timer for leaving the group

Join: LeaveAll Specify the timer for leaving all groups

. smi\_xmrp\_set\_leave\_timer

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- ← vr\_id Virtual Router ID <0-255>
- ← *protocol* Protocol type string {gmrp|mmrp}
- ← *ifname* Interface name
- ← *timer\_value* Timer value in centiseconds <1-0xffffffff>

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_ERR\_GMRP\_NOCONFIG\_ONPORT NSM\_BRIDGE\_ERR\_GENERAL

# 4.1.2.36 s\_int32\_t smi\_xmrp\_set\_registration\_type (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol, char \* ifname, char \* reg\_type)

Sets the GMRP/MMRP port's multicast group registration type to one of the following.

fixed: The multicast groups currently registered on the switch will remain on the port, but subsequent new registrations or de-registrations based on timers do not affect the port.

forbidden: All registered multicast groups are de-registered, and prevents further multicast registration on the port.

normal: Set multicast group registration and de-registration to dynamic.

restricted: Set to restricted registration.

NOTE: GMRP cannot be enabled if IGMP snooping is enabled. smi\_xmrp\_set\_registration\_type

- $\leftarrow$  azg Pointer to the SMI client global structure
- ← *vr\_id* Virtual Router ID <0-255>
- ← *protocol* Protocol type string {gmrp|mmrp}
- *← ifname* Interface name
- ← *reg\_type* Registration types {fixed|forbidden|normal|restricted}

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_GENERAL NSM\_BRIDGE\_ERR\_GMRP\_NOCONFIG NSM\_GMRP\_ERR\_GMRP\_NOT\_CFG\_ON\_VLAN NSM\_GMRP\_ERR\_VLAN\_NOT\_CFG\_ON\_PORT NSM\_ERR\_GMRP\_NOCONFIG\_ONPORT NSM\_GMRP\_ERR\_GMRP\_NOT\_CFG\_ON\_PORT\_VLAN NSM\_GMRP\_ERR\_GMRP\_GLOBAL\_CFG\_PORT

4.1.2.37 s\_int32\_t smi\_xmrp\_show\_bridge\_configuration (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol, char \* bridge\_name, struct list \* brConfList, int(\*)(struct list \*brConfList) funPointer)

Show GMRP/MMRP bridge configuration. smi\_xmrp\_show\_bridge\_configuration

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- ← *vr\_id* Virtual Router ID <0-255>
- ← *protocol* Protocol type string {gmrp|mmrp}
- ← *bridge\_name* Bridge name
- → brConfList Pointer to linked list of structure xmrpBridgeConfig
- ← funPointer Callback function pointer

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_NOTFOUND NSM\_BRIDGE\_ERR\_GMRP\_NOCONFIG CLI\_ERROR

4.1.2.38 s\_int32\_t smi\_xmrp\_show\_finite\_state\_machine (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol, char \* bridge\_name, struct list \* fsmStateList, int(\*)(struct list \*fsmStateList) funPointer)

Show GMRP/MMRP finite state machine details. smi\_xmrp\_show\_finite\_state\_machine\_details

- ← azg Pointer to the SMI client global structure
- $\leftarrow vr \ id \ \text{Virtual Router ID} < 0-255 >$
- ← *protocol* Protocol type string {gmrp|mmrp}

- ← *bridge\_name* Bridge name
- → fsmStateList Pointer to linked list of structure xmrpFsmState
- ← *funPointer* Callback function pointer

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_NOTFOUND NSM\_BRIDGE\_ERR\_GMRP\_NOCONFIG CLI\_ERROR

4.1.2.39 s\_int32\_t smi\_xmrp\_show\_per\_vlan\_statistics (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol, char \* bridge\_name, u\_int16\_t vlan\_id, struct xmrpVlanStats \* vlanStats)

Show various GMRP/MMRP related statistics of particular VLAN. smi\_xmrp\_show\_per\_vlan\_statistics

#### **Parameters:**

- ← azg Pointer to the SMI client global structure
- ← vr\_id Virtual Router ID <0-255>
- ← *protocol* Protocol type string {gmrp|mmrp}
- ← *vlan\_id* VLAN number <1-4094>
- ← *bridge\_name* Bridge name <1-32>
- $\rightarrow$  *vlanStats* Pointer to structure xmrpVlanStats

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_GENERAL RESULT\_ERROR

4.1.2.40 s\_int32\_t smi\_xmrp\_show\_times (struct smiclient\_globals \* azg, u\_int32\_t vr\_id, char \* protocol, char \* ifname, struct xmrpTimers \* timerData)

Show GMRP/MMRP configured timer values. smi\_xmrp\_show\_timers

- ← azg Pointer to the SMI client global structure
- $\leftarrow vr_{id}$  Virtual Router ID < 0-255>
- ← *protocol* Protocol type string {gmrp|mmrp}
- ← *ifname* Interface name

 $\rightarrow$  timerData Pointer to xmrpTimers structure

#### **Returns:**

0 on success, otherwise one of the following error codes NSM\_BRIDGE\_ERR\_NOTFOUND NSM\_BRIDGE\_ERR\_GMRP\_NOCONFIG CLI\_ERROR

### 4.2 smi\_gmrp\_msg.h File Reference

Defines data structures used by gmrp/mmrp SMI APIs. #include "pal.h"
#include "message.h"
#include "thread.h"
#include "network.h"
#include "log.h"
#include "tlv.h"
#include "syslog.h"
#include <sys/types.h>
#include "pal\_types.h"
#include "pal\_socket.h"
#include "prefix.h"
#include "smi server.h"

#### **Data Structures**

- struct xmrpTimers
- struct xmrpBridgeConfig
- struct xmrpFsmState
- struct xmrpVlanStats
- struct xmrpCommonList
- struct gmrp\_msg\_

#### **Defines**

- #define SMI\_GMRP\_EVENT 1
- #define SMI GMRP TIMER 2
- #define SMI\_GMRP\_CLI 3
- #define **SMI\_GMRP\_PACKET** 4
- #define SMI GMRP ALL 5
- #define GMRP\_DEBUG\_EVENT 0x01
- #define GMRP\_DEBUG\_CLI 0x02
- #define GMRP\_DEBUG\_TIMER 0x04
- #define **GMRP\_DEBUG\_PACKET** 0x08
- #define SMI\_BRIDGE\_MIN\_VAL 1
- #define **SMI\_BRIDGE\_MAX\_VAL** 32
- #define SMI\_VLAN\_ID\_MIN\_VAL 2
- #define SMI\_VLAN\_ID\_MAX\_VAL 4094
- #define SMI\_XMRP\_TIMER\_MIN 1
- #define **SMI\_XMRP\_TIMER\_MAX** 0xffffffff

- #define XMRP\_TOTAL\_ATTR\_EVENTS 7
- #define SMI\_GMRP\_CTYPE\_VR\_ID 0
- #define SMI\_GMRP\_CTYPE\_PROTOCOL 1
- #define SMI\_GMRP\_CTYPE\_BRIDGE\_NAME 2
- #define SMI\_GMRP\_CTYPE\_VLAN\_ID 3
- #define SMI GMRP CTYPE IFNAME 4
- #define SMI GMRP CTYPE TIMER VALUE 5
- #define SMI\_GMRP\_CTYPE\_REG\_TYPE 6
- #define SMI\_XMRP\_CTYPE\_SHOW\_TIMERS 7
- #define SMI\_XMRP\_CTYPE\_SHOW\_BRIDGE 8
- #define SMI\_XMRP\_CTYPE\_SHOW\_FSM 9
- #define SMI\_XMRP\_CTYPE\_SHOW\_STATS 10
- #define SMI\_GMRP\_CTYPE\_IS\_GMRP\_EN 11
- #define SMI\_GMRP\_CTYPE\_TIMER\_TYPE 12
- #define SMI\_GMRP\_CTYPE\_IS\_P2P 13
- #define SMI\_GMRP\_CTYPE\_PERIODIC\_TIMER\_STATUS 14
- #define SMI GMRP CTYPE REGIS TYPE 15
- #define SMI GMRP CTYPE FORWARD ALL 16
- #define SMI\_GMRP\_CTYPE\_EXTENDED\_FILTERING\_FLAG 17
- #define SMI\_GMRP\_CTYPE\_SMI\_DEBUG 18
- #define SMI\_GMRP\_CTYPE\_EXTENDED\_1 31

#### **Typedefs**

• typedef struct gmrp\_msg\_ gmrp\_msg

#### **Enumerations**

```
    enum smi_gmrp_garp_timers {
    SMI_GMRP_GARP_JOIN_TIMER = 0, SMI_GMRP_GARP_LEAVE_-
TIMER = 1, SMI_GMRP_GARP_LEAVE_ALL_TIMER = 2, SMI_-
GMRP_GARP_LEAVE_CONF_TIMER = 3,
    SMI_GMRP_GARP_LEAVEALL_CONF_TIMER = 4, SMI_GMRP_-
GARP_PERIODIC_TIMER = 5, SMI_GMRP_GARP_MAX_TIMERS = 6
}
```

#### **Functions**

- int **smi\_parse\_gmrp** (u\_char \*\*pnt, u\_int16\_t \*size, struct smi\_msg\_header \*header, void \*arg, SMI\_CALLBACK callback)
- int smi encode gmrp (u char \*\*pnt, u int16 t \*size, gmrp msg \*msg)
- int smi\_decode\_gmrp (u\_char \*\*pnt, u\_int16\_t \*size, gmrp\_msg \*msg)

#### **4.2.1** Detailed Description

Defines data structures used by gmrp/mmrp SMI APIs.

# **Index**

gmrp_msg_, 5	smi_xmrp_enable_per_port_per vlan, 27
smi_gmrp.h, 11	smi_xmrp_enable_per_vlan, 28
smi_gmrp_debug_set, 17	smi_xmrp_get_extended_filtering
smi_gmrp_debug_unset, 17	flag, 28
smi_gmrp_show_debugging, 17	smi_xmrp_get_forward_all_flag, 29
smi_is_gmrp_enabled, 18	smi_xmrp_get_registration_type, 29
smi_mmrp_disable_periodic	smi_xmrp_get_timer, 30
timer_global, 18	smi_xmrp_set_join_timer, 30
smi_mmrp_disable_periodic	smi_xmrp_set_leave_all_timer, 31
timer_per_port, 19	smi_xmrp_set_leave_timer, 31
smi_mmrp_enable_periodic_timer	smi_xmrp_set_registration_type, 32
global, 19	smi_xmrp_show_bridge
smi_mmrp_enable_periodic_timer	configuration, 33
per_port, 20	smi_xmrp_show_finite_state
smi_mmrp_get_periodic_timer	machine, 33
per_port_status, 20	smi_xmrp_show_per_vlan
smi_mmrp_set_if_mode_to_p2p, 20	statistics, 34
smi_mmrp_unset_if_mode_to_p2p,	smi_xmrp_show_times, 34
smi_mmrp_verify_p2p_mode_set,	smi_gmrp_debug_set
21	smi_gmrp.h, 17
smi_xmrp_clear_all_vlan_statistics,	smi_gmrp_debug_unset
21	smi_gmrp.h, 17
smi_xmrp_clear_per_vlan_statistics,	smi_gmrp_msg.h, 36
22	smi_gmrp_show_debugging
smi_xmrp_disable_all_port, 22	smi_gmrp.h, 17
smi_xmrp_disable_extended	smi_is_gmrp_enabled
filtering, 23	smi_gmrp.h, 18
smi_xmrp_disable_forward_all, 23	smi_mmrp_disable_periodic_timer
smi_xmrp_disable_global, 23	global
smi_xmrp_disable_per_port, 24	smi_gmrp.h, 18
smi_xmrp_disable_per_port_per	smi_mmrp_disable_periodic_timer_per_
vlan, 24	port
smi_xmrp_disable_per_vlan, 25	smi_gmrp.h, 19
smi_xmrp_enable_all_port, 25	smi_mmrp_enable_periodic_timer
smi_xmrp_enable_extended	global
filtering, 26	smi_gmrp.h, 19
smi_xmrp_enable_forward_all, 26	smi_mmrp_enable_periodic_timer_per
smi_xmrp_enable_global, 26	port
smi_xmrp_enable_per_port, 27	smi_gmrp.h, 20

INDEX 39

smi_mmrp_get_periodic_timer_per	smi_gmrp.h, 30
port_status	smi_xmrp_set_leave_all_timer
smi_gmrp.h, 20	smi_gmrp.h, 31
smi_mmrp_set_if_mode_to_p2p	smi_xmrp_set_leave_timer
smi_gmrp.h, 20	smi_gmrp.h, 31
smi_mmrp_unset_if_mode_to_p2p	smi_xmrp_set_registration_type
smi_gmrp.h, 21	smi_gmrp.h, 32
smi_mmrp_verify_p2p_mode_set	smi_xmrp_show_bridge_configuration
smi_gmrp.h, 21	smi_gmrp.h, 33
smi_xmrp_clear_all_vlan_statistics	smi_xmrp_show_finite_state_machine
smi_gmrp.h, 21	smi_gmrp.h, 33
smi_xmrp_clear_per_vlan_statistics	smi_xmrp_show_per_vlan_statistics
smi_gmrp.h, 22	smi_gmrp.h, 34
smi_xmrp_disable_all_port	smi_xmrp_show_times
smi_gmrp.h, 22	smi_gmrp.h, 34
smi_xmrp_disable_extended_filtering	
smi_gmrp.h, 23	xmrpBridgeConfig, 6
smi_xmrp_disable_forward_all	xmrpCommonList, 7
smi_gmrp.h, 23	xmrpFsmState, 8
smi_xmrp_disable_global	xmrpTimers, 9
smi_gmrp.h, 23	xmrpVlanStats, 10
smi_xmrp_disable_per_port	
smi_gmrp.h, 24	
smi_xmrp_disable_per_port_per_vlan	
smi_gmrp.h, 24	
smi_xmrp_disable_per_vlan	
smi_gmrp.h, 25	
smi_xmrp_enable_all_port	
smi_gmrp.h, 25	
smi_xmrp_enable_extended_filtering	
smi_gmrp.h, 26	
smi_xmrp_enable_forward_all	
smi_gmrp.h, 26	
smi_xmrp_enable_global	
smi_gmrp.h, 26	
smi_xmrp_enable_per_port	
smi_gmrp.h, 27	
smi_xmrp_enable_per_port_per_vlan	
smi_gmrp.h, 27	
smi_xmrp_enable_per_vlan	
smi_gmrp.h, 28	
smi_xmrp_get_extended_filtering_flag	
smi_gmrp.h, 28	
smi_xmrp_get_forward_all_flag	
smi_gmrp.h, 29	
smi_xmrp_get_registration_type	
smi_gmrp.h, 29	
smi_xmrp_get_timer	
smi_gmrp.h, 30	
smi_xmrp_set_join_timer	