

## Chapter 10

# Generic Attribute Registration Protocol (GARP)

Introduction .....	10-2
GARP Operations .....	10-2
Overview of Generic VLAN Registration Protocol (GVRP) .....	10-5
Configuring GARP .....	10-6
Command Reference .....	10-8
disable garp .....	10-8
disable garp debug .....	10-10
disable garp gip .....	10-11
enable garp .....	10-12
enable garp debug .....	10-13
enable garp gip .....	10-15
reset garp .....	10-16
set garp port .....	10-17
set garp timer .....	10-19
show garp .....	10-21
show garp counter .....	10-23
show garp db .....	10-26
show garp debug .....	10-27
show garp gip .....	10-28
show garp machine .....	10-29

## Introduction

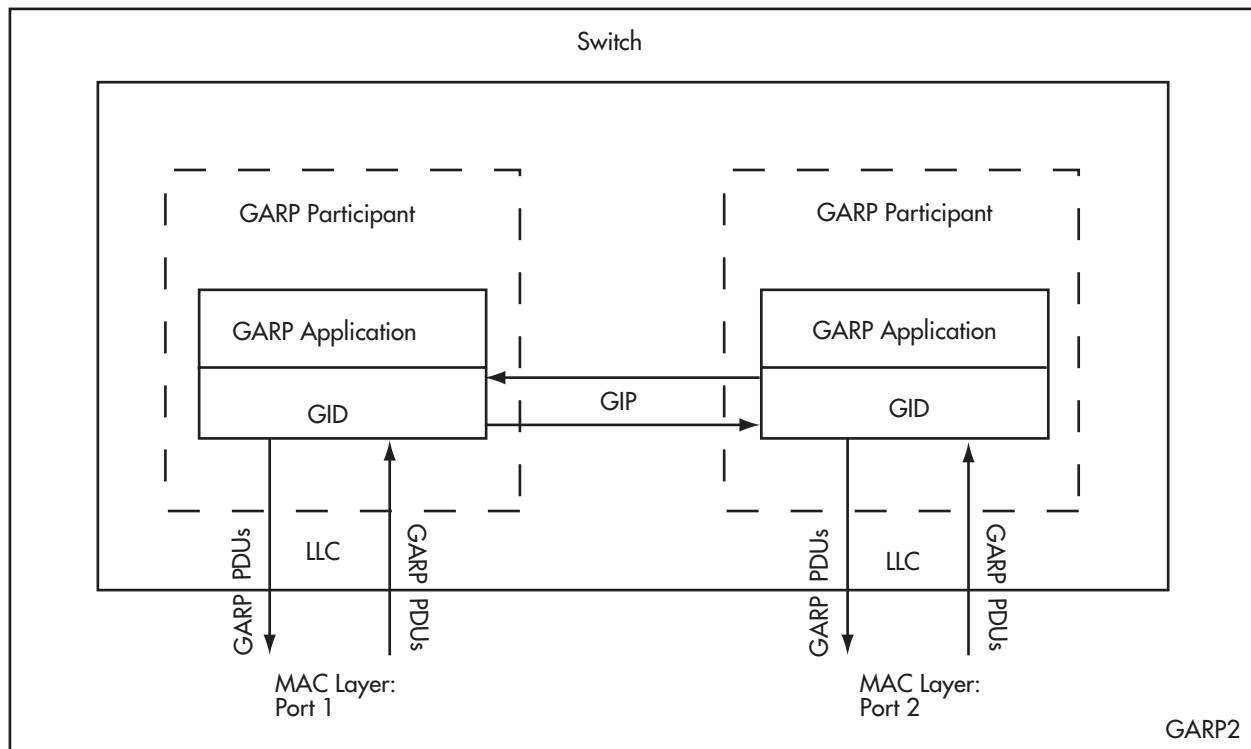
The purpose of the Generic Attribute Registration Protocol (GARP) is to provide a generic framework whereby devices in a bridged LAN, such as end stations and switches, can register and de-register *attribute* values, such as VLAN Identifiers, with each other. In doing so, the attributes are propagated to devices in the bridged LAN, and these devices form a “reachability” tree that is a subset of an active topology. For a bridged LAN, the active topology is normally that created and maintained by the Spanning Tree Protocol (STP).

GARP defines the architecture, rules of operation, state machines, and variables for the registration and de-registration of attribute values. By itself, GARP is not directly used by devices in a bridged LAN; GARP applications specify what an attribute represents and so perform meaningful actions. For example, the GARP VLAN Registration Protocol (VVRP) application distributes dynamic filter entries for VLAN membership among the Forwarding Databases of VLAN-aware switches.

## GARP Operations

A GARP participant in a switch or an end station consists of a GARP application component and a GARP Information Declaration (GID) component associated with each port of the switch. One such GARP participant exists per port, per GARP application. The following figure shows GARP architecture.

Figure 10-1: GARP architecture



The propagation of information between GARP participants for the same application in a switch is carried out by the GARP Information Propagation (GIP) component. Protocol exchanges take place between GARP Participants

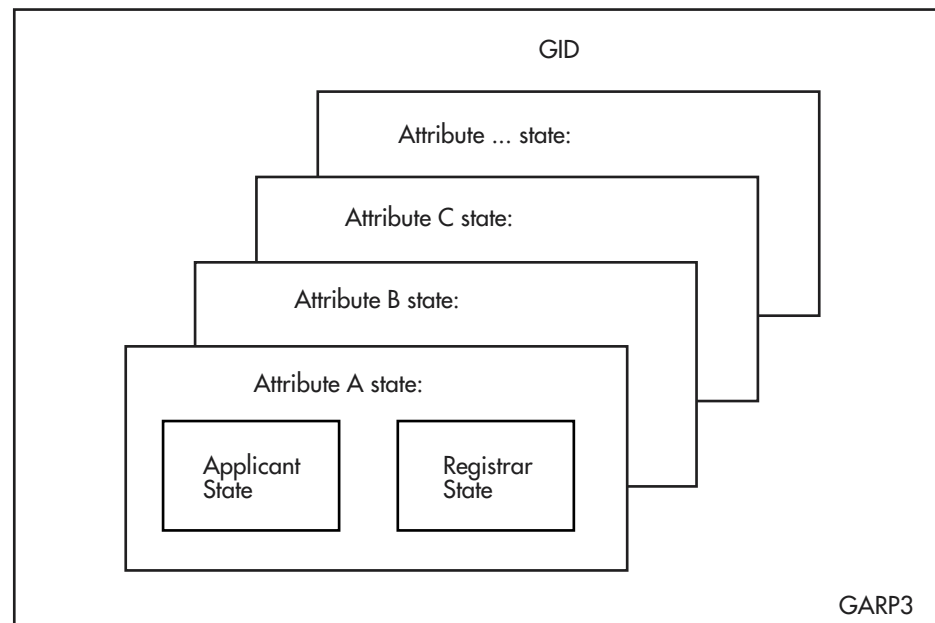
by means of LLC Type 1 services, using the group MAC address and PDU format defined for the GARP application concerned.

Every instance of a GARP application includes a database to store the values of the attributes. Within GARP, attributes are mapped to GID indexes.

The GARP application component of the GARP Participant is responsible for defining the semantics associated with the parameter values and operators received in GARP PDUs, and for generating GARP PDUs for transmission. The application uses the GID component and the state machines associated with the operation of GID in order to control its protocol interactions.

An instance of GID consists of the set of state machines that define the current registration and declaration state of all *attribute* values associated with the GARP Participant. Separate state machines exist for the applicant and registrar. The following figure shows GID architecture.

Figure 10-2: GID architecture



GARP registers and de-registers *attribute* values through GARP messages sent at the GID level. A GARP participant that wants to make a declaration (an applicant registering an *attribute* value) sends a JoinIn or JoinEmpty message. An applicant that wants to withdraw a declaration (de-registering an *attribute* value) sends a LeaveEmpty or LeaveIn message. Following the de-registration of an *attribute* value, the applicant sends a number of Empty messages. The purpose of the Empty message is to prompt other applicants to send JoinIn/JoinEmpty messages. For the GARP protocol to be resilient against multiple lost messages, a LeaveAll message is available. Timers are used in the state machines to generate events and control state transitions.

The job of the applicant is to ensure that:

- this participant's declarations are registered by other participants' registrars
- other participants have a chance to re-declare (rejoin) after anyone withdraws a declaration (leaves).

Therefore, the applicant is looking after the interests of all would-be participants, which allows the registrar to be very simple.

The job of the registrar is to record whether an attribute is registered, in the process of being de-registered, or is not registered for an instance of GID.

The Applicant Administrative Control parameter controls the applicant state machine. This parameter determines whether the applicant state machine participates in GARP protocol exchanges. The default is for the applicant to participate in exchanges.

The Registrar Administrative Control parameter controls the registrar state machine. This parameter determines whether the registrar state machine listens to incoming GARP messages. The default is for the registrar to listen to incoming messages.

The propagation of information between GARP Participants for the same Application in a switch is carried out by the GIP component. The operation of GIP is dependent upon STP being enabled on a port, as only ports in the STP Forwarding state are eligible for membership to the GIP connected ring. Ports in the GIP connected ring propagate GID Join and Leave requests to notify each other of attribute registrations and de-registrations. The operation of GIP allows ports in the switch to share information between themselves and the LANs/end stations to which the ports are connected.

If a port enters the STP Forwarding state and the GARP application that the port belongs to is enabled, then the port is added to the GIP connected ring for the GARP application. All attributes registered by other ports in the GIP connected ring are propagated to the recently connected port. All attributes registered by the recently connected port are propagated to all other ports in the GIP connected ring.

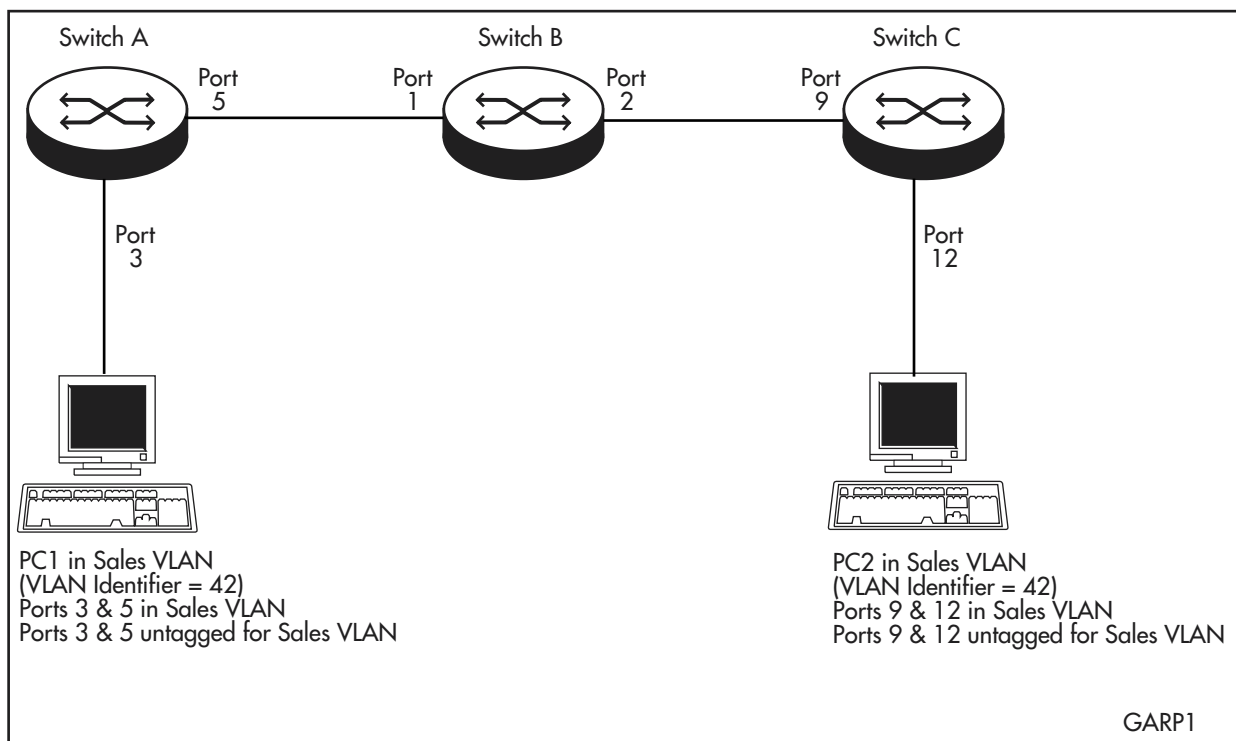
Similarly, if a port leaves the STP Forwarding state and the GARP application that the port belongs to is enabled, then the port is removed from the GIP connected ring for the GARP application. Prior to removal, GID leave requests are propagated to all other ports in the GIP connected ring if the port to be removed has previously registered an attribute and no other port in the GIP connected ring has registered that attribute. The operations of GIP can be enabled or disabled by user command.

## Overview of Generic VLAN Registration Protocol (GVRP)

The Generic VLAN Registration Protocol (GVRP) allows a LAN device to signal other neighbouring devices that it wishes to receive packets for one or more VLANs. The main purpose of GVRP is to allow switches to automatically discover some of the VLAN information that would otherwise have to be manually configured in each switch. This is achieved by using GARP to propagate VLAN Identifier attributes across a bridged LAN. GVRP can also be run by network servers. These servers are usually configured to join several VLANs, and then signal the network switches of the VLANs they want to join.

The following figure shows a network example using GVRP. Switches A, B and C are VLAN-aware. Switch B has not been configured to route packets between Switches A and C.

Figure 10-3: Network example using GVRP



Without GVRP operating, end stations PC1 and PC2 can communicate to each other only if a network administrator has configured ports 1 and 2 on Switch B to be in the same VLAN. For a large network, such manual configuration is required on all switches with the correct allocation of ports into VLANs on all switches.

By enabling GVRP on Switches A, B and C, the Sales VLAN Identifier (42) is propagated to Switch B on ports 1 and 2. These ports register the VLAN Identifier attribute and a VLAN named `gvrp42` with VLAN Identifier 42 is created on Switch B. Ports 1 and 2 on Switch B is added into the `gvrp42` VLAN as tagged ports. Thus, Switch B is now able to forward Sales VLAN traffic from Switch A to Switch C and vice-versa, and end stations PC1 and PC2 can communicate to each other.

GVRP creates dynamic VLANs in switches that do not already have a VLAN with the VLAN Identifier propagated by GARP.

These VLANs cannot be destroyed by the [destroy vlan command on page 8-77 of Chapter 8, Switching](#). Ports assigned to the dynamic VLANs cannot be modified by the user.

Static VLAN attribute registrations and de-registrations occur every time a port is added to or deleted from a VLAN, or a VLAN is added to or deleted from an STP.

Dynamic VLAN attribute registrations and de-registrations are triggered by receiving GARP PDUs and by the timeout of the LeaveAll timer.

Creating an instance of an STP also creates an instance of a GVRP application. Destroying the STP instance destroys the instance of the GVRP application. Because a default STP exists in the switch, a default instance of GVRP exists in the switch. Whenever a port is added to an STP (either when a VLAN that contains ports is added to or removed from an STP, or when a port is added to or removed from a VLAN), then a GID instance is created representing the port in the GVRP instance. The GID instances in a GVRP instance mirror the ports in the underlying STP instance.

## Configuring GARP

To use GARP, a GARP application must be created. The Layer 3 switch has one GARP application - the GARP VLAN Registration Protocol (GVRP). The purpose of GVRP is to propagate VLAN information between VLAN-aware switches. GVRP creates dynamic VLANs when GARP is enabled on the switch.

GVRP is disabled by default. The **garp** parameter in the GARP commands specifies the GARP application. Because one GARP application has been implemented, GVRP must be specified.

To enable or disable packet reception and transmission operation of a GARP application, use the commands:

```
enable garp=gvrp [stp={stp-name|all}]
disable garp=gvrp [stp={stp-name|all}]
```

These commands have precedence over the per-port mode setting.

To set port parameters of a GARP application, use the command:

```
set garp port={port-list|all} [mode={none|normal}]
[showdebug={on|off}] [stp={stp-name|all}]
```

When ports are added to a GARP application, the new ports are in normal mode and **showdebug** is on.

Timers generate events and control state transitions. Timers measure time in units of deciseconds (tenths of a second). To set timeout values used by all timers in a GARP application, use the commands:

```
set garp timer [jointime=4..60] [leavetime=12..120]
[leavealltime=96..6000] [stp={stp-name|all}]

set garp=gvrp timer default [stp={stp-name|all}]
```

To clear dynamically learned information from a GARP application, use the command:

```
reset garp=gvrp [stp={stp-name|all}]
```

This has the side-effect of destroying dynamically created VLANs made during the operation of the GVRP instance. However, these VLANs may be re-created immediately following the command if the switch receives GVRP packets.

To display information for all GARP applications, use the commands:

```
show garp
```

```
show garp=gvrp [stp={stp-name|all}]
```

To display GVRP information for all STPs, use either of the following commands:

```
show garp=gvrp
```

```
show garp=gvrp stp=all
```

## Command Reference

---

This section describes the commands available to configure GARP in the Layer 3 switch.

Because GMRP may also be implemented, command handlers require the explicit declaration of the GARP application. This means that GARP=GVRP must be specified instead of GARP with no value for all commands other than the **show garp** commands.

Additional parameters, parameter options, and parameter descriptions specific to the Rapier *i* Series switch are noted in the command description section.

The shortest valid command is denoted by capital letters in the Syntax section. See [“Conventions” on page lxvi of About this Software Reference](#) for details of the conventions used to describe command syntax. See [Appendix A, Messages](#) for a complete list of messages and their meanings.

### disable garp

---

**Syntax** `DISable GARP=GVRP [STP={name|CIST|ALL}]`

where *name* is an STP name or an MST instance (MSTI):

- An STP name is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0-9), the underscore, and hyphen. The *name* cannot be **all**.
- An MSTP MSTI instance name comprises the letters "MSTI" followed by a number from 1 to 4094.

**Description** This command disables the packet reception and transmission operation of a GARP application. The default setting is for all GARP applications to be disabled. This command has precedence over the per-port mode setting.

Disabling a GARP application purges all non-static attribute registrations from the GARP application. Existing static attribute registrations are moved down in the database for the GARP application to form a contiguous sequence, similarly for the GID state machines. All ports in the GIP connected ring are disconnected. Ports in a disabled GARP application are not re-connected into the GIP connected ring if the STP status for the port changes. The LeaveAll timer is stopped on all ports in the GARP application.

Disabling a GARP application does not affect port movements that affect the GARP application. For example, if the GVRP instance of the default STP is disabled, and a VLAN that has untagged member ports is deleted from the default STP, then member ports of the VLAN are also deleted from the GVRP instance of the default STP.

The **garp** parameter specifies the GARP application. Because one GARP application has been implemented, GVRP must be specified.

The **stp** parameter specifies which GVRP instance is affected by this command. Each spanning tree instance has its own instance of GVRP. The instance specified in this parameter could be an STP or an MSTP instance. If **cist** is



specified, it is the MSTP CIST. If an MSTP instance is specified, such as MSTI20, it is an MSTP MSTI. If **all** is specified, then all GVRP instances are affected.

**Examples** To disable all GVRP instances, use either of the commands:

```
dis garp=gvrp
```

```
dis garp=gvrp stp=all
```

To disable the GVRP instance of the accounting STP, use the command:

```
dis garp=gvrp stp=accounting
```

**Related Commands** [enable garp](#)  
[show garp](#)  
[show garp counter](#)

## disable garp debug

**Syntax** DISable GARP=GVRP DEBug [=ALL | MSG | PKT | STAtE]  
[STP={*name* | CIST | ALL}]

where *name* is an STP name or an MST instance (MSTI):

- An STP name is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0-9), the underscore, and hyphen. The *name* cannot be **all**.
- An MSTP MSTI instance name comprises the letters "MSTI" followed by a number from 1 to 4094.

**Description** This command disables the debugging option of a GARP application. The default setting is for the debugging option of a GARP application to be disabled. This command is effective on disabled GARP applications. By specifying the **showdebug** parameter on the **set garp port** command, the debugging output can be switched on or off on a per-port, per STP instance basis.

On a Rapier *i* switch only, by specifying the **showdebug** parameter on the **set garp port** command the output of debugging can be switched on or off on a per-port, per STP instance basis.

The **garp** parameter specifies the GARP application. Because one GARP application has been implemented, GVRP must be specified.

The **debug** parameter specifies which debugging options to disable. The value of this parameter is a single item. The following table explains debugging options.

Option	Description
msg	Decoded display of received and transmitted GARP packets.
pkt	Raw ASCII display of received and transmitted GARP packets.
state	GID attribute state machine transitions.
all	All debugging options.

The **stp** parameter specifies which GVRP instance is affected by this command. Each spanning tree instance has its own instance of GVRP. The instance specified in this parameter could be an STP or an MSTP instance. If **cist** is specified, it is the MSTP CIST. If an MSTP instance is specified, such as MSTI20, it is an MSTP MSTI. If **all** is specified, then all GVRP instances will be affected.

**Examples** To disable packet debugging of the instance of GVRP belonging to the sales STP, use the command:

```
dis garp=gvrp deb=pkt stp=sales
```

**Related Commands** [enable garp debug](#)  
[set garp port](#)  
[show garp debug](#)

## disable garp gip

---

**Syntax** `DISable GARP=GVRP GIP [STP={name|CIST|ALL}]`

where *name* is an STP name or an MST instance (MSTI):

- An STP name is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0-9), the underscore, and hyphen. The *name* cannot be **all**.
- An MSTP MSTI instance name comprises the letters "MSTI" followed by a number from 1 to 4094.

**Description** This command disables the operation of GIP for the GARP application. By disabling GIP, attribute registrations and de-registrations processed on a port are not propagated to other ports in the GIP connected ring. All ports in the GIP connected ring are removed from the ring as a result of this command. The default state is for GIP operation to be enabled for a GARP application.

The **garp** parameter specifies the GARP application. Because one GARP application has been implemented, GVRP must be specified.

The **stp** parameter specifies which GVRP instance is affected by this command. Each spanning tree instance has its own instance of GVRP. The instance specified in this parameter could be an STP or an MSTP instance. If **cist** is specified, it is the MSTP CIST. If an MSTP instance is specified, such as MSTI20, it is an MSTP MSTI. If **all** is specified, then all GVRP instances are affected.

**Examples** To disable GIP operation for all GVRP instances, use either of the commands:

```
dis garp=gvrp gip
dis garp=gvrp gip stp=all
```

To disable GIP operation for the GVRP instance of the accounting STP, use the command:

```
dis garp=gvrp gip stp=accounting
```

**Related Commands** [enable garp gip](#)  
[show garp](#)

## enable garp

---

**Syntax** `ENABle GARP=GVRP [STP={name|CIST|ALL}]`

where *name* is an STP name or an MST instance (MSTI):

- An STP name is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0-9), the underscore, and hyphen. The *name* cannot be **all**.
- An MSTP MSTI instance name comprises the letters "MSTI" followed by a number from 1 to 4094.

**Description** This command enables the packet reception and transmission operation of a GARP application. The default setting is for all GARP applications to be disabled. This command has precedence over the per-port mode setting.

Enabling a GARP application re-connects all ports of the GARP application in the STP Forwarding state into the GIP connected ring. The LeaveAll timer is started on all ports in the GARP application.

The **garp** parameter specifies the GARP application. Because one GARP application has been implemented, GVRP must be specified.

The **stp** parameter specifies which GVRP instance is affected by this command. Each spanning tree instance has its own instance of GVRP. The instance specified in this parameter could be an STP or an MSTP instance. If **cist** is specified, it is the MSTP CIST. If an MSTP instance is specified, such as MSTI20, it is an MSTP MSTI. If **all** is specified, then all GVRP instances are affected.

**Examples** To enable all GVRP instances, use either of the commands:

```
ena garp=gvrp
ena garp=gvrp stp=all
```

To enable the GVRP instance of the accounting STP, use the command:

```
ena garp=gvrp stp=accounting
```

**Related Commands** [disable garp](#)  
[show garp](#)  
[show garp counter](#)

## enable garp debug

**Syntax** `ENABle GARP=GVRP DEBUg [=ALL | MSG | PKT | STAtE]  
[OUTput=CONsole TIMEOut=NONE | decimal]  
[STP={name | CIST | ALL}]`

where *name* is an STP name or an MST instance (MSTI):

- An STP name is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0-9), the underscore, and hyphen. The *name* cannot be **all**.
- An MSTP MSTI instance name comprises the letters "MSTI" followed by a number from 1 to 4094.

**Description** This command enables the debugging option of a GARP application. The default setting is for the debugging option of a GARP application to be disabled. Be aware that enabling debug could flood the receiving Telnet session or asynchronous port with raw data.

This command is effective on disabled GARP applications. By specifying the **showdebug** parameter in the **set garp port** command, the output of debugging can be switched on or off on a per-port, per STP instance basis.

On the Rapier *i* switch only, by specifying the **showdebug** parameter in the **set garp port** command, the output of debugging can be switched on or off on a per-port, per STP instance basis.

The **garp** parameter specifies the GARP application. Because one GARP application has been implemented, GVRP must be specified.

The **debug** parameter specifies which debugging option to enable. The value of this parameter is a single item. The following table explains debugging options:

Option	Description
msg	Decoded display of received and transmitted GARP packets.
pkt	Raw ASCII display of received and transmitted GARP packets.
state	GID attribute state machine transitions.
all	All debugging options.

The **output** parameter set to **console** specifies that the debugging information produced is sent to the console. By default the debugging data is sent to the port that received the **enable garp debug** command. Use the **output** option if the **enable garp debug** command is used in a script, since a script is not received on a port.

The **timeout** parameter specifies the number of seconds for which debugging is enabled on the specified ports. This value overrides any previous GARP debugging timeout values for these ports, even if they were specified for other debugging modes. If **timeout** is not specified, the time is the most recent timeout value previously set with the **enable garp debug** command, or **none** if it has not been previously set.

The **stp** parameter specifies which GVRP instance is affected by this command. Each STP has its own instance of GVRP. If **all** is specified, then all GVRP instances are affected.

**Examples** To enable packet debugging of the instance of GVRP belonging to the sales STP, use the command:

```
ena garp=gvrp deb=pkt stp=sales
```

**Related Commands** [disable garp debug](#)  
[set garp port](#)  
[show garp debug](#)

## enable garp gip

---

**Syntax** `ENABle GARP=GVRP GIP [STP={name|CIST|ALL}]`

where *name* is an STP name or an MST instance (MSTI):

- An STP name is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0-9), the underscore, and hyphen. The *name* cannot be **all**.
- An MSTP MSTI instance name comprises the letters "MSTI" followed by a number from 1 to 4094.

**Description** This command enables the operation of GIP for the GARP application. By enabling GIP, attribute registrations and de-registrations processed on a port are propagated to other ports in the GIP connected ring. All ports in the STP Forwarding state are added to the GIP connected ring as a result of this command. The default state is for GIP operation to be enabled for a GARP application. GIP operation is dependent upon having STP enabled on switch ports.

The **garp** parameter specifies the GARP application. Because one GARP application has been implemented, GVRP must be specified.

The **stp** parameter specifies which GVRP instance is affected by this command. Each STP has its own instance of GVRP. If **all** is specified, then all GVRP instances are affected.

**Examples** To enable GIP operation for all GVRP instances, use either of the commands:

```
ena garp=gvrp gip
ena garp=gvrp gip stp=all
```

To enable GIP operation for the GVRP instance of the accounting STP, use the command:

```
ena garp=gvrp gip stp=accounting
```

**Related Commands** [disable garp gip](#)  
[show garp](#)

## reset garp

---

**Syntax** RESET GARP=GVRP [STP={*name* | CIST | ALL}]

where *name* is an STP name or an MST instance (MSTI):

- An STP name is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0-9), the underscore, and hyphen. The *name* cannot be **all**.
- An MSTP MSTI instance name comprises the letters "MSTI" followed by a number from 1 to 4094.

**Description** This command clears dynamically learnt information from a GARP application. Counters for the GARP application are initialised to zero. During the processing of this command, all ports are disconnected from the GIP connected ring. The ports are re-connected into the GIP connected ring if the GARP application is enabled and the ports are in the STP Forwarding state. This ensures attributes are propagated correctly through the GIP connected ring.

The **garp** parameter specifies the GARP application. Because one GARP application has been implemented, GVRP must be specified.

The **stp** parameter specifies which GVRP instance is affected by this command. Each STP has its own instance of GVRP. If **all** is specified, then all GVRP instances are affected.

**Examples** To reset all GVRP instances, use either of the commands:

```
reset garp=gvrp
reset garp=gvrp stp=all
```

To reset the GVRP instance of the accounting STP, use the command:

```
reset garp=gvrp stp=accounting
```

**Related Commands** [show garp](#)  
[show garp counter](#)



## set garp port

**Syntax** SET GARP=GVRP Port={*port-list* | ALL} [MODE={NONE | NORMaL}]  
[SHOWdebug={ON | OFF}] [STP={*name* | CIST | ALL}]

where:

- *name* is an STP name or an MST instance (MSTI):
  - An STP name is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0-9), the underscore, and hyphen.
  - An MSTP MSTI instance name comprises the letters "MSTI" followed by a number from 1 to 4094.

**Description** This command sets the port parameters of a GARP application. Whenever ports are added to a GARP application (typically by using the [add vlan port](#) or [add stp vlan](#) command) the new ports are in normal mode and **showdebug** is on.

The **garp** parameter specifies the GARP application. Because one GARP application has been implemented, GVRP must be specified.

The **port** parameter specifies the ports that belong to the GARP application. If **all** is specified, then all ports belonging to the GARP application are affected by this command. If the GARP application has no ports and **all** is specified, then a message is displayed. If the command applies to an individual GARP application, then the ports specified must all belong to that GARP application. If the command applies to all GARP applications of a certain type, for example all GVRP applications, then the command succeeds on the GARP applications to which the specified ports belong.

On the Rapier *i* switch only, if the port-list supplied does not match all the STP instances perfectly, the command still succeeds as a whole.

The **mode** parameter specifies the mode of operation of the applicant and registrar state machines. If **none** is specified, then the GARP application does not listen to or send GARP messages in a GARP PDU. If **normal** is specified, then the GARP application listens to and sends GARP messages in GARP PDUs. The default is **normal**.

The **showdebug** parameter specifies whether debugging information is displayed for the port if debugging is enabled on the GARP application. If **on** is specified, then debugging information is displayed as per the debug mode setting for the GARP application (see the **show garp debug** command). If **off** is specified, then no debugging information is displayed for the port. It is recommended that this command be entered first with **showdebug** set to **off** for all ports in the GARP application before debugging is enabled on the GARP application. This is because of the large volume of debugging information that can be generated. The default is **on**.

On the Rapier *i* switch only, the **stp** parameter specifies which GVRP instance is affected by this command. Each STP has its own instance of GVRP. The default is **all**. Because a port can belong to more than one STP instance, a port can belong to more than one instance of a GVRP. If **all** is specified for a port, then all GVRP instances that belong to the port are affected.

**Examples** To set the mode of operation to **none** for ports 3 and 6 of the GVRP instance belonging to the accounting STP, use either of the commands:

```
set garp=gvrp po=3,6 mod=none stp=accounting
```

```
set garp=gvrp po=3,6 mod=none
```

To not display debugging information for port 8 of the respective GVRP instance, use the command:

```
set garp=gvrp po=8 show=off
```

**Related Commands**

- [disable garp debug](#)
- [enable garp debug](#)
- [show garp](#)
- [show garp debug](#)

## set garp timer

**Syntax** SET GARP=GVRP TImEr [JOInTime=4..60] [LEAVETime=12..120]  
[LEAVEAllTime=96..6000] [STP={*name*|CIST|ALL}]

SET GARP=GVRP TImEr DEFault [STP={*name*|CIST|ALL}]

where *name* is an STP name or an MST instance (MSTI):

- An STP name is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0-9), the underscore, and hyphen. The *name* cannot be **all**.
- An MSTP MSTI instance name comprises the letters "MSTI" followed by a number from 1 to 4094.

**Description** This command sets the timeout values used by all timers in a GARP application.

The **garp** parameter specifies the GARP application. Because one GARP application has been implemented, GVRP must be specified.

The **jointime** parameter sets the duration of the Join Period timer for the GARP applications in deciseconds. The timer is used to control the interval between opportunities to transmit GARP PDUs. Each port in the GARP application has a Join Period timer. The default is 5 deciseconds (0.5 seconds).

The **leavetime** parameter sets the duration of the Leave Period timer for the GARP applications in deciseconds. The timer controls how long the registrar state machine waits in the LV state before entering the MT state, for example, how quickly a registrar de-registers an attribute from the registered IN state. Each port in the GARP application has a single Leave Period timer. Although a Leave Period timer is strictly required on a per-attribute basis, this implementation of GARP uses a Leave Timer that expires four times to timeout the leave time period. This granularity allows multiple Leave Period durations to be timed out using a single timer on each port belonging to the GARP application. The default is 15 deciseconds (1.5 seconds).

The **leavealltime** parameter sets the duration of the LeaveAll Period timer for the GARP application in deciseconds. The timer controls the frequency with which the LeaveAll state machine generates a LeaveAll message in a GARP PDU. Each port in the GARP application has a LeaveAll timer. The default is 120 deciseconds (12 seconds).

The Join Period, Leave Period and LeaveAll Period durations are interrelated. The two criteria for setting the timers are:

$$\text{leavetime} \geq \text{jointime} * 3$$

derived from IEEE standard 802.1D, and

$$\text{leavealltime} \geq \text{leavetime} * 8$$

which states that the LeaveAll Period must be significantly greater than the Leave Period, as per the IEEE standard.

The **default** parameter sets the Join Period, Leave Period, and LeaveAll Period timers to their default values. The **default** parameter cannot be specified in conjunction with any timer duration parameter.

The **stp** parameter specifies which GVRP instance is affected by this command. Each STP has its own instance of GVRP. If **all** is specified, then all GVRP instances are affected.

**Examples** To set the Join Period timer to 1 second, Leave Period timer to 3.5 seconds and the LeaveAll Period timer to 36 seconds for all GVRP applications, use either of the commands:

```
set garp=gvrp tim join=10 leavet=35 leavea=360
set garp=gvrp tim join=10 leavet=35 leavea=360 stp=all
```

To set the timers to their defaults for all GVRP applications, use either of the commands:

```
set garp=gvrp tim def
set garp=gvrp tim def stp=all
```

To set the Join Period timer to 0.4 seconds for the GVRP instance belonging to the marketing STP, use the command:

```
set garp=gvrp tim join=4 stp=marketing
```

**Related Commands** [show garp](#)

## show garp

**Syntax** `SHOW GARP=GVRP [STP={name|CIST|ALL}]`

where *name* is an STP name or an MST instance (MSTI):

- An STP name is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0-9), the underscore, and hyphen. The *name* cannot be **all**.
- An MSTP MSTI instance name comprises the letters "MSTI" followed by a number from 1 to 4094.

**Description** This command displays information about the GARP application ([Figure 10-4](#), [Table 10-1 on page 10-22](#)).

Figure 10-4: Example output from the **show garp** command

```
GARP Information
-----
Size of Database ..... 100

GARP Application ..... GVRP
STP ..... fred
  Status ..... ON
  GIP Operation ..... OFF
  JoinTime ..... 5
  LeaveTime ..... 15
  LeaveAllTime ..... 120
  Hold Time ..... 2

Port Information:
  Mode NONE ..... -
  Mode NORMAL ..... 10-24
  Show Debug OFF ..... -
  Show Debug ON ..... 10-24

Attribute Information:
  VLAN Identifier(s) ..... 11 (v11)

GARP Application ..... GVRP
STP ..... default
  Status ..... ON
  GIP Operation ..... OFF
  JoinTime ..... 5
  LeaveTime ..... 15
  LeaveAllTime ..... 120
  Hold Time ..... 2

Port Information:
  Mode NONE ..... 2
  Mode NORMAL ..... 1,3-9
  Show Debug OFF ..... -
  Show Debug ON ..... 1-9

Attribute Information:
  VLAN Identifier(s) ..... 1 (default)
-----
```

Table 10-1: Parameters in output of the **show garp** command

Parameter	Meaning
Size of Database	Maximum size of the internal database belonging to each GARP application. This database matches attribute values to GID indexes used by GARP. This parameter is fixed.
GARP Application	Identifies the GARP application, i.e. "GVRP".
STP	Present if the GARP application is GVRP and identifies the STP that owns this instance of GVRP.
Status	Whether the GARP application is on or off.
GIP Operation	Whether the GIP operation for the GARP application is on or off.
JoinTime	Duration of the Join Period timer in deciseconds. This parameter can be set by Management.
LeaveTime	Duration of the Leave Period timer in deciseconds. This parameter can be set by Management.
LeaveAllTime	Duration of the LeaveAll Period timer in deciseconds. This parameter can be set by Management.
Hold Time	Duration of the Hold timer in deciseconds. This parameter is fixed. The Hold timer prevents successive opportunities to transmit occurring immediately.
Port Information	Whether port settings for the GARP application is mode or show debug.
Mode	Whether the mode for ports in the GARP application is normal or none. A list of ports is printed if the mode setting is applicable on the port.
Show Debug	Whether the show debug setting is on or off for ports in the GARP application. A list of ports is printed if the mode setting is applicable on the port.
Attribute Information	Values of the attributes presently registered by the GARP application. The attributes are specific to each type of GARP application. If no attributes are registered, "None" is displayed.
VLAN Identifier(s)	Value of the GVRP VID Attribute, a VLAN Identifier. The name of the VLAN is shown in brackets.

**Examples** To display information for all GARP applications, use the command:

```
sh garp
```

To display GVRP information for all STPs, use either of the following commands:

```
sh garp=gvrp
```

```
sh garp=gvrp stp=all
```

**Related Commands**

- [disable garp](#)
- [enable garp](#)
- [reset garp](#)
- [set garp port](#)
- [set garp timer](#)

## show garp counter

**Syntax** SHow GARP COUnTer

SHow GARP=GVRP COUnTer [STP={*name*|CIST|ALL}]

where *name* is an STP name or an MST instance (MSTI):

- An STP *name* is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0-9), the underscore, and hyphen. The *name* cannot be **all**.
- An MSTP MSTI instance name comprises the letters "MSTI" followed by a number from 1 to 4094.

**Description** This command displays information about the GARP packet and message counters (Figure 10-5, Table 10-2 on page 10-23).

Figure 10-5: Example output from the **show garp counter** command

GARP Application: GVRP		STP: default	
Receive:		Transmit:	
Total GARP Packets	1	Total GARP Packets	5
Invalid GARP Packets	0		
Discarded:			
GARP Disabled	0	GARP Disabled	24
Port Not Listening	0	Port Not Sending	0
Invalid Port	0		
Invalid Protocol	0		
Invalid Format	0		
Database Full	0		
GARP Messages:			
LeaveAll	0	LeaveAll	24
JoinEmpty	1	JoinEmpty	157
JoinIn	0	JoinIn	112
LeaveEmpty	0	LeaveEmpty	31
LeaveIn	0	LeaveIn	0
Empty	0	Empty	9
Bad Message	0		
Bad Attribute	0		

Table 10-2: Parameters in output of the **show garp counter** command

Parameter	Meaning
GARP Application	Identifies the GARP application, i.e. "GVRP".
STP	Present if the GARP application is GVRP; identifies the STP that owns this instance of GVRP.
Receive: Total GARP Packets	Total number of GARP packets (PDUs) received by this GARP application.
Transmit: Total GARP Packets	Total number of GARP packets (PDUs) transmitted by this GARP application.
Receive: Invalid GARP Packets	Number of invalid GARP packets (PDUs) received by this GARP application.
Receive Discarded: GARP Disabled	Number of received GARP packets (PDUs) discarded because the GARP application was disabled.

Table 10-2: Parameters in output of the **show garp counter** command (Continued)

Parameter	Meaning
Transmit Discarded: GARP Disabled	Number of GARP packets (PDUs) discarded because the GARP application was disabled. This counter is incremented when ports are added to or deleted from the GARP application arising from port movements in the underlying VLAN or STP.
Receive Discarded: Port Not Listening	Number of GARP packets (PDUs) discarded because the port that the packets were received on was not listening, i.e. mode=none has been set on the port.
Transmit Discarded: Port Not Sending	Number of GARP packets (PDUs) discarded because the port that the packets were to be transmitted on was not sending, i.e. mode=none has been set on the port.
Receive Discarded: Invalid Port	Number of GARP packets (PDUs) discarded because the port that the packet was received on does not belong to the GARP application.
Receive Discarded: Invalid Protocol	Number of GARP packets (PDUs) discarded because the GARP PDU contained an invalid protocol.
Receive Discarded: Invalid Format	Number of GARP packets (PDUs) discarded because the format of the GARP PDU was not recognised.
Receive Discarded: Database Full	Number of GARP packets (PDUs) discarded because the database for the GARP application was full, i.e. maximum number of attributes for the GARP application in use.
Receive GARP Messages: LeaveAll	Number of GARP LeaveAll messages received by the GARP application.
Transmit: GARP Messages: LeaveAll	Number of GARP LeaveAll messages transmitted by the GARP application.
Receive GARP Messages: JoinEmpty	Total number of GARP JoinEmpty messages received for all attributes in the GARP application.
Transmit GARP Messages: JoinEmpty	Total number of GARP JoinEmpty messages transmitted for all attributes in the GARP application.
Receive GARP Messages: JoinIn	Total number of GARP JoinIn messages received for all attributes in the GARP application.
Transmit GARP Messages: JoinIn	Total number of GARP JoinIn messages transmitted for all attributes in the GARP application.
Receive GARP Messages: LeaveEmpty	Total number of GARP LeaveEmpty messages received for all attributes in the GARP application.
Transmit GARP Messages: LeaveEmpty	Total number of GARP LeaveEmpty messages transmitted for all attributes in the GARP application.
Receive GARP Messages: LeaveIn	Total number of GARP LeaveIn messages received for all attributes in the GARP application.
Transmit GARP Messages: LeaveIn	Total number of GARP LeaveIn messages transmitted for all attributes in the GARP application.
Receive GARP Messages: Empty	Total number of GARP Empty messages received for all attributes in the GARP application.
Transmit GARP Messages: Empty	Total number of GARP Empty messages transmitted for all attributes in the GARP application.
Receive GARP Messages: Bad Message	Number of GARP messages that had an invalid Attribute Type value, an invalid Attribute Length value or an invalid Attribute Event value.



Table 10-2: Parameters in output of the **show garp counter** command (Continued)

Parameter	Meaning
Receive GARP Messages: Bad Attribute	Number of GARP messages that had an invalid Attribute Value value.

**Examples** To display information for all GARP application counters, use the command:

```
sh garp cou
```

**Related Commands** [reset garp](#)

## show garp db

**Syntax** SHOW GARP [=GVRP] DB

**Description** This command displays the internal database for the GARP application (Figure 10-6, Table 10-3). Each attribute is represented by a GID index within the GARP application.

Figure 10-6: Example output from the **show garp db** command

GARP Application: GVRP			STP: default		
GID index	Attribute	Used	GID index	Attribute	Used
0	1	Yes	1	11	No
2	42	Yes			

Table 10-3: Parameters in output of the **show garp db** command

Parameter	Meaning
GARP Application	Identifies the GARP application, i.e. "GVRP".
STP	Present if the GARP application is GVRP; identifies the STP that owns this instance of GVRP.
GID index	Value of the GID index corresponding to the attribute. GID indexes begin at 0. If the GARP application has no attributes presently registered, "No attributes have been registered" is displayed.
Attribute	Value of the attribute.
Used	Whether the GID index is currently being used by a port in the GARP application. Specifically, whether the applicant and registrar state machine for the GID index are in a non-initialised state, i.e. not in {Vo, Mt} state.

**Examples** To display the database for all GARP applications, use the command:

```
sh garp db
```

**Related Commands** [show garp machine](#)

## show garp debug

**Syntax** SHOW GARP DEBUg

SHOW GARP=GVRP DEBUg [STP={*name*|CIST|ALL}]

where *name* is an STP name or an MST instance (MSTI):

- An STP name is a string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0-9), the underscore, and hyphen. The *name* cannot be **all**.
- An MSTP MSTI instance name comprises the letters "MSTI" followed by a number from 1 to 4094.

**Description** This command displays debugging information for the GARP application (Figure 10-7, Table 10-4).

Figure 10-7: Example output from the **show garp debug** command

GARP Application: GVRP	STP: default	
Enabled Debug Modes	Output	Timeout
-----	-----	-----
PKT	16	None
-----	-----	-----

Table 10-4: Parameters in output of the **show garp debug** command

Parameter	Meaning
GARP Application	Identifies the GARP application, for example, GVRP.
STP	Present if the GARP application is GVRP; identifies the STP that owns this instance of GVRP.
Enabled Debug Modes	Whether the debugging option for the GARP application is <b>msg</b> , <b>pkt</b> , <b>state</b> , or <b>none</b> .
Output	Output device for the GARP application; shown if a debug mode is presently enabled.
Timeout	Length of time in seconds that the debugging options for the GARP application are enabled; shown if a debug mode is presently enabled. If no timeout has been set or the timeout is set to <b>none</b> , "None" is displayed.

**Examples** To display debugging information for all GARP applications, use the command:

```
sh garp deb
```

**Related Commands** [disable garp debug](#)  
[enable garp debug](#)

## show garp gip

**Syntax** SHow GARP [=GVRP] GIP

**Description** This command displays the GIP connected ring for the GARP application (Figure 10-8, Table 10-5).

Figure 10-8: Example output from the **show garp gip** command

```
GARP Application: GVRP          STP: default
Connected Ring:
24 -> 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 ->
10 -> 11 -> 12 -> 13 -> 14 -> 15 -> 16 -> 17 -> 18 -> 19 ->
20 -> 21 -> 22 -> 23 -> 24
```

Table 10-5: Parameters in output of the **show garp gip** command

Parameter	Meaning
GARP Application	Identifies the GARP application, i.e. "GVRP".
STP	Present if the GARP application is GVRP; identifies the STP that owns this instance of GVRP.
Connected Ring	Ring of connected ports. Ports presently in the STP Forwarding state are eligible for membership to the GIP connected ring. If no ports exist in the GIP connected ring, "No ports are connected" is displayed. If the GARP application has no ports, "No ports have been assigned" is displayed.

**Examples** To display the GIP connected ring for all GARP applications, use the command:

```
sh garp gip
```

**Related Commands** [disable garp](#)  
[enable garp](#)  
[reset garp](#)

## show garp machine

**Syntax** SHow GARP [=GVRP] MACHine

**Description** This command displays the GID state machines for the GARP application (Figure 10-9, Table 10-6). The output is shown on a per-GID index basis; each attribute is represented by a GID index within the GARP application.

Figure 10-9: Example output from the **show garp machine** command

GARP Application: GVRP				STP: default											
GID	Port	App	Reg	Port	App	Reg	Port	App	Reg	Port	App	Reg	Port	App	Reg
0	1	Qa	Fix	2	Qa	Fix	3	Qa	Fix	4	Qa	Fix	5	Qa	Fix
	5	Qa	Fix	6	Qa	Fix	7	Qa	Fix	8	Qa	Fix	9	Qa	Fix
	9	Vo	Mt	10	Vo	Mt	11	Vo	Mt	12	Vo	Mt	13	Vo	Mt
	13	Vo	Mt	14	Vo	Mt	15	Vo	Mt	16	Vo	Mt	17	Vo	Mt
	17	Qa	Fix	18	Qa	Fix	19	Qa	Fix	20	Qa	Fix	21	Qa	Fix
	21	Qa	Fix	22	Qa	Fix	23	Qa	Fix	24	Qa	Fix	25	Qa	Fix
1	1	Vo	Mt	2	Vo	Mt	3	Vo	Mt	4	Vo	Mt	5	Vo	Mt
	5	Vo	Mt	6	Vo	Mt	7	Vo	Mt	8	Vo	Mt	9	Vo	Mt
	9	Vo	Mt	10	Vo	Mt	11	Vo	Mt	12	Vo	Mt	13	Vo	Mt
	13	Vo	Mt	14	Vo	Mt	15	Vo	Mt	16	Vo	Mt	17	Vo	Mt
	17	Vo	Mt	18	Vo	Mt	19	Vo	Mt	20	Vo	Mt	21	Vo	Mt
	21	Vo	Mt	22	Vo	Mt	23	Vo	Mt	24	Vo	Mt	25	Vo	Mt

Table 10-6: Parameters in output of the **show garp machine** command

Parameter	Meaning
GARP Application	Identifies the GARP application, i.e. "GVRP".
STP	Present if the GARP application is GVRP; identifies the STP that owns this instance of GVRP.
App	Applicant state machine for the GID index on that particular port.  Participant management states are:  Vo - Very Anxious Observer Ao - Anxious Observer Qo - Quiet Observer Lo - Leaving Observer Vp - Very Anxious Passive Member Ap - Anxious Passive Member Qp - Quiet Passive Member Va - Very Anxious Active Member Aa - Anxious Active Member Qa - Quiet Active Member La - Leaving Active Member

Table 10-6: Parameters in output of the **show garp machine** command (Continued)

Parameter	Meaning
App ( <i>Continued</i> )	Non-participant management states are: Von Very Anxious Observer Aon Anxious Observer Qon Quiet Observer Lon Leaving Observer Vpn Very Anxious Passive Member Apn Anxious Passive Member Qpn Quiet Passive Member Van Very Anxious Active Member Aan Anxious Active Member Qan Quiet Active Member Lan Leaving Active Member The initialised state for the applicant is Vo.
Reg	Registrar state machine for the GID index on that particular port. States are: Mt Empty Lv3 Leaving substate 3 (final Leaving substate) Lv2 Leaving substate 2 Lv1 Leaving substate 1 Lv Leaving substate (initial Leaving substate) In In Fix Registration Fixed For Registration Forbidden The initialised state for the registrar is Mt.

**Examples** To display GID state machines for all GARP applications, use the command:

```
sh garp mach
```

**Related Commands** [show garp db](#)