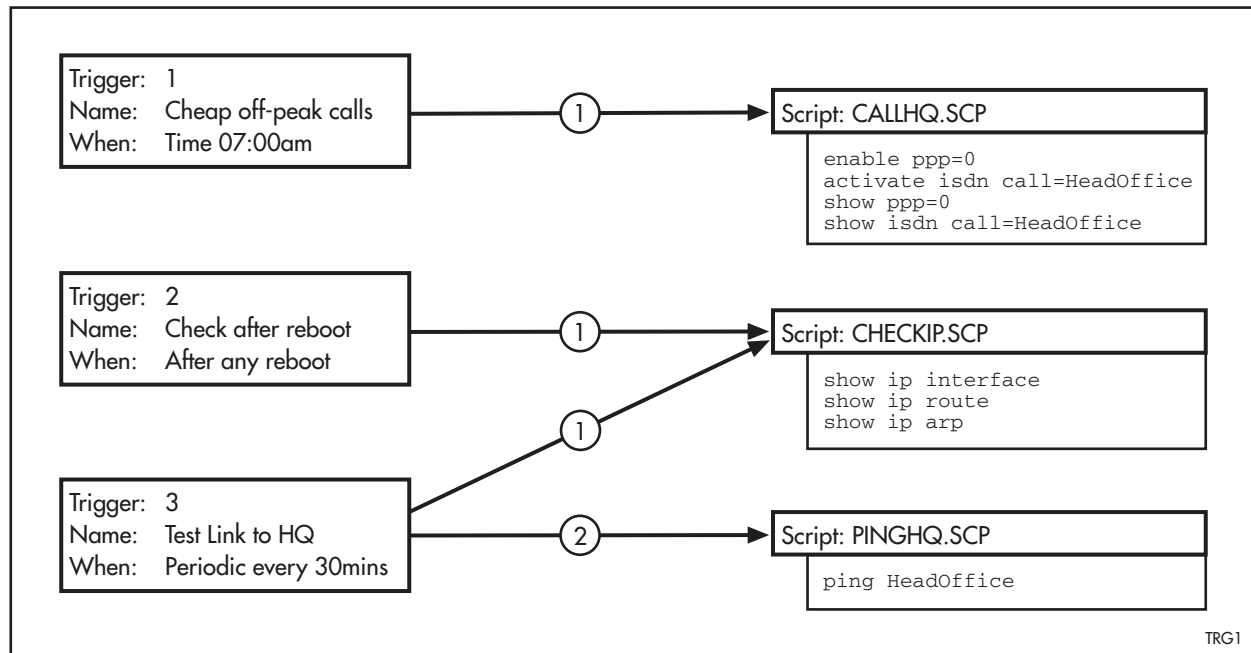**Chapter 59**

Trigger Facility

Introduction

The Trigger facility provides a powerful mechanism for automatic and timed management of the router by automating the execution of router commands in response to certain events. For example, triggers can be configured to activate and deactivate an ISDN call at specified times, or to collect diagnostic information after a router reboots.

Some interface and port types mentioned in this chapter may not be supported on your router. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the Hardware Reference.

A *trigger* is an ordered sequence of scripts to be executed when a certain event occurs. A *script* is a sequence of router commands stored as a plaintext file in the router's file subsystem in flash memory. Each trigger may reference multiple scripts and any script may be used by any trigger. Various types of triggers are supported, each activated in a different way (Figure 59-1).

Figure 59-1: Triggers respond to events by performing a sequence of predefined scripts

When a trigger is activated, the scripts associated with it are executed in sequence by the router. The trigger passes script parameters to the script that depend on the trigger type, and in the case of module triggers, the event that activates the trigger. These parameters are available to the script that is executed. Global script parameters that contain date, time, system name, and serial number of the router are also available to scripts. See "Script Parameters" on page 58-5 of Chapter 58, Scripting for more information about passing parameters to scripts. The output from the scripts is passed to the Logging facility, and can be displayed with the **show log** command on page 60-37 of Chapter 60, Logging Facility, or forwarded to another router.

Defining Triggers

Three main groups of parameters define each trigger. Trigger type parameters define the kind of event that activates a trigger. Some modules have their own module type triggers, with additional module-specific defining parameters described in the chapter for the individual module. There are also a group of general trigger parameters that can be used with any trigger type.

To create a trigger, use the **create trigger** command on page 59-8.

To change a trigger, use the **set trigger** command on page 59-18.

Each trigger can be assigned a descriptive name and up to five scripts can be executed. See Chapter 58, Scripting for more information about creating scripts. To add a script to any position in the sequence, use the **add trigger** command on page 59-7.

To delete scripts from a trigger, use the **delete trigger** command on page 59-15.

To destroy a trigger, use the **destroy trigger** command on page 59-15.

By default, triggers are enabled when created. A trigger can be disabled by using the **disable trigger** command on page 59-16.

To activate a trigger, first enable the trigger module if it is not already enabled, then enable the trigger. Both of these are done using the **enable trigger** command on page 59-17. You must have the Trigger module enabled before triggers will activate.

A trigger can be explicitly activated (triggered) by using the **activate trigger** command on page 59-6 regardless of whether it is currently enabled.

The **show trigger** command on page 59-24displays summary information about all triggers or details about a particular one.

Configuration Example

The example in this section shows how to initiate ISDN calls during off-peak periods.

Some interface and port types mentioned in this example may not be supported on your router. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC) is installed. For more information, see the Hardware Reference.

The example assumes that the charging regime for regional ISDN calls is such that calls initiated between 8 a.m. and 8 p.m. are charged at a rate twice that of calls between 8 p.m. and 8 a.m, and that the rate for the entire call is based on the rate at the time the call is initiated, regardless of how long the call lasts. For a network link that is often busy during the working day, it may be cheaper to activate calls before 8 a.m. and keep the link up until 6 p.m. rather than make numerous calls after 8 a.m.

To automatically activate ISDN calls during off-peak charging periods

1. **Create the ISDN call.**

    Create the ISDN call, specifying any required options, using the command:

    ```
    add isdn call=cheap number=42 precedence=out
    ```

2. **Create a PPP interface to use the ISDN call.**

    Create a PPP interface to use the call, and set the idle time to the default of 36000 seconds (the number of seconds between 8am and 6pm). This ensures that the call stays up once it is activated at 8am.

    ```
    create ppp=0 over=isdn-cheap idle=36000
    ```

3. **Create a script to activate the ISDN call.**

    Create a script that explicitly sets the idle time of the PPP interface to 36000 seconds (to keep the call up all day), and activates the ISDN call:

    ```
    add script=acheap.scp text="set ppp=0 idle=36000"

    add script=acheap.scp text="activate isdn call=cheap"
    ```

4. **Create a trigger to use the script.**

    Create a time trigger to activate at 7:59am and execute the script:

    ```
    enable trigger

    create trigger=1 time=07:59 days=weekdays
        script=acheap.scp name="enable off-peak calls"
        repeat=forever
    ```

5. **Create a script to deactivate the ISDN call.**

    Create a script that explicitly sets the idle time of the PPP interface to the default of 60 seconds (to activate the call when there is traffic during the night), and deactivates the ISDN call:

    ```
    add script=dcheap.scp text="set ppp=0 idle=60"
    ```

    Note that the ISDN call is not explicitly deactivated in case traffic is being transmitted over the link. The call automatically deactivates when there has been no traffic for 60 seconds.

6. **Create a trigger to use the script.**

    Create a time trigger to activate at 6pm and execute the script:

    ```
    create trigger=2 time=18:00 days=weekdays
        script=dcheap.scp name="disable off-peak calls"
        repeat=forever
    ```

7. **Save the dynamic configuration**

    Save the modified dynamic configuration as the script file OFFPEAK.CFG
    and make OFFPEAK.CFG the boot script:

    ```
    create config=offpeak.cfg
    ```

    ```
    set config=offpeak.cfg
    ```

Command Reference

This section describes the commands to configure and manage the trigger facility in the router. The trigger facility requires that the router's internal clock be set correctly. See Chapter 4, Configuring and Monitoring the System for descriptions of the commands required to configure the router's internal clock.

Some interface and port types mentioned in this chapter may not be supported on your router. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC) is installed. For more information, see the Hardware Reference.

The shortest valid command is denoted by capital letters in the Syntax section. See "Conventions" on page lxv of About this Software Reference in the front of this manual for details of the conventions used to describe command syntax. See Appendix A, Messages for a complete list of messages and their meanings.

activate trigger

Syntax   ACTivate TRIGger=*trigger-id*

where *trigger-id* is a number from 1 to 250

Description   This command immediately activates a specific trigger even if it has been disabled with the **disable trigger** command. The scripts associated with the trigger are executed even if the **test** option is set. Normally, a **test** trigger generates only log entries when it triggers and does not invoke scripts.

The **trigger** parameter specifies the number of the trigger to activate. The specified trigger must already exist.

Triggers activated manually do not have their repeat counts decremented or their "*last triggered*" time updated, and do not result in updates to the "*time/periodic triggers today*" counters.

Examples   To activate trigger number 8, use the command:

```
act trig=8
```

Related Commands   create trigger
disable trigger
enable trigger
show trigger

add trigger

Syntax ADD TRIGger=*trigger-id* SCript=*filename*... [NUMber=*index*]

where:

- *trigger-id* is a number from 1 to 250.

- *filename* is a file name in the format [device]:`filename.ext`. The *.ext* must be .scp or .cfg. Valid characters are lowercase and uppercase letters, digits (0–9), and the characters ~ ' ! @ # $ % ^ & ( ) _ - { }. Invalid characters are * + = " | \ [ ] ; : ? / , < >. Wildcards are not allowed. The *device* is optional, and specifies the physical memory device on which the file is stored, which is flash. If *device* is specified, it must be separated from the rest of the file name by a colon.

- *index* is a number from 1 to *n*+1, where *n* is the number of scripts already assigned to the trigger.

Description This command adds a script to a trigger so that the script is executed when the trigger is activated.

The **trigger** parameter specifies the number of the trigger to which the script is to be added. The specified trigger must already exist.

The **script** parameter specifies the name of the script to be added. The **script** parameter may be repeated up to five times in one command in order to add up to five scripts (in the order specified) at once.

The **number** parameter specifies the position in the script list where the script is to be added. If **number** is specified, the new script occupies this position and all following scripts are pushed down one position. The default position is at the end of the list of scripts.

Examples To add scripts snapshot.scp and callhq.scp to trigger 1 at position 3, use the command:

```
add trig=1 sc=snapshot.scp sc=callhq.scp num=3
```

Related Commands create trigger
delete trigger
disable trigger
enable trigger
set trigger
show trigger

create trigger

Syntax   CREate TRIGger=*trigger-id* CPU=*value* [DIrection={UP|DOWN|
ANY}] [AFTer=*hh:mm*] [BEfore=*hh:mm*] [{DAte=*date*|
DAYs=*day-list}*] [NAMe=*name*] [REPeat={Yes|No|ONCe|
FORever|*count*}] [SCript=*filename*...] [STAte={ENAbled|
DIsabled}] [TEST={YES|NO|ON|OFF|True|False}]

CREate TRIGger=*trigger-id* FIREwall={ALL|DOSattack|
FRAgattack|HOStscan|PORtscan|SESSION|SIPAutomax|
SMTPATTACK|SMUrfattack|SYNattack|TCPattack}
[MODE={STArt|END|BOTH}] [AFTer=*hh:mm*] [BEfore=*hh:mm*]
[{DAte=*date*|DAYs=*day-list*}] [NAMe=*name*] [REPeat={Yes|
No|ONCe|FORever|*count*}] [SCript=*filename*...]
[STAte={ENAbled|DIsabled}] [TEST={YES|NO|ON|OFF|True|
False}]

CREate TRIGger=*trigger-id* INTerface=*interface* EVent={UP|
DOWN|FAIL|ANY} [CIRCuit=*miox-circuit*] [CP={APPLE|ATCP|
BCP|CCP|DCP|DNCP|IPcp|IPXcp|LCP}] [DLCi=*dlci*]]
[AFTer=*hh:mm*] [BEfore=*hh:mm*] [{DAte=*date*|
DAYs=*day-list*}] [NAMe=*name*] [REPeat={Yes|No|ONCe|
FORever|*count*}] [SCript=*filename*...] [STAte={ENAbled|
DIsabled}] [TEST={YES|NO|ON|OFF|True|False}]

CREate TRIGger=*trigger-id* MEMory=*value* [DIrection={UP|
DOWN|ANY}] [AFTer=*hh:mm*] [BEfore=*hh:mm*] [{DAte=*date*|
DAYs=*day-list*}] [NAMe=*name*] [REPeat={Yes|No|ONCe|
FORever|*count*}] [SCript=*filename*...] [STAte={ENAbled|
DIsabled}] [TEST={YES|NO|ON|OFF|True|False}]

CREate TRIGger=*trigger-id* MODule=*module* EVent=*event*
[*module-parameters*...] [AFTer=*hh:mm*] [BEfore=*hh:mm*]
[{DAte=*date*|DAYs=*day-list*}] [NAMe=*name*] [REPeat={Yes|
No|ONCe|FORever|*count*}] [SCript=*filename*...]
[STAte={ENAbled|DIsabled}] [TEST={YES|NO|ON|OFF|True|
False}]

CREate TRIGger=*trigger-id* MODule=SWItch PORT=*port*
EVent={LINKDOWN|LINKUP} [AFTer=*hh:mm*] [BEfore=*hh:mm*]
[{DAte=*date*|DAYs=*day-list*}] [NAMe=*name*] [REPeat={Yes|
No|ONCe|FORever|*count*}] [SCript=*filename*...]
[STAte={ENAbled|DIsabled}] [TEST={YES|NO|ON|OFF|True|
False}]

CREate TRIGger=*trigger-id* PERiodic=*minutes* [AFTer=*hh:mm*]
[BEfore=*hh:mm*] [{DAte=*date*|DAYs=*day-list*}] [NAMe=*name*]
[REPeat={Yes|No|ONCe|FORever|*count*}]
[SCript=*filename*...] [STAte={ENAbled|DIsabled}]
[TEST={YES|NO|ON|OFF|True|False}]

CREate TRIGger=*trigger-id* REBoot={REStart|CRASH|ALL}
[AFTer=*hh:mm*] [BEfore=*hh:mm*] [{DAte=*date*|
DAYs=*day-list*}] [NAMe=*name*] [REPeat={Yes|No|ONCe|
FORever|*count*}] [SCript=*filename*...] [STAte={ENAbled|
DIsabled}] [TEST={YES|NO|ON|OFF|True|False}]

```
CREate TRIGger=trigger-id TIme=hh:mm [{DAte=date|
   DAYs=day-list}] [SCript=filename...] [NAMe=name]
   [REPeat={Yes|No|ONCe|FORever|count}] [STAte={ENAbled|
   DIsabled}] [TEST={YES|NO|ON|OFF|True|False}]
```

where:

- *trigger-id* is a number from 1 to 250.

- *module* is the name of a router module as given in "Module identifiers, display names, and descriptions" on page B-2 of Appendix B, Reference Tables.

- *event* is an event defined in the specified module

- *module-parameters* are one or more other required or optional parameters for the specified module and event, whose syntax is defined in the specified module.

- *value* is a number from 1 to 100.

- *interface* is a valid interface name.

- *miox-circuit* is an alphanumeric string 1 to 15 characters long.

- *dlci* is a Frame Relay Data Link Connection Identifier (DLCI), from 0 to 1023.

- *minutes* is a number from 1 to 1439.

- *hh:mm* is a time in hours and minutes.

- *date* is a date in the format dd-mmm-yyyy, where *mmm* is the first three letters of the month name.

- *day-list* is one or more keywords for the day of the week.

- *filename* is a file name in the format [device]:filename.ext. The *.ext* must be .scp or .cfg. Valid characters are uppercase and lowercase letters, digits (0–9), and the characters ~ ' ! @ # $ % ^ & ( ) _ - { }. Invalid characters are * + = " | \ [ ] ; : ? / , < >. Wildcards are not allowed. The *device* is optional and specifies the physical memory device on which the file is stored, which is flash. If specified, it must be separated from the rest of the file name by a colon.

- *name* is a string 1 to 40 characters long. If the string contains spaces, it must be in double quotes.

- *count* is a number from 1 to 4294967294 (2^{32}-2).

Description   This command creates a new trigger and defines events and conditions that activate it. It requires a user with security officer privilege when the router is in security mode. Different trigger types are supported—CPU triggers, firewall triggers, link triggers, memory triggers, module triggers, periodic triggers, reboot triggers, and time triggers. Some of these have additional trigger-specific parameters. The **event** parameter and other parameters for module triggers are described in the chapter for the module that supports them. General trigger parameters can be specified for all trigger types: **after**, **before**, **days** or **date**, **script**, **name**, **repeat**, **state**, and **test**.

When a trigger is activated, it passes parameters to the scripts that it executes. The global script parameters containing the date (%D), time (%T), system name (%N) and serial number (%S) of the router are passed to scripts executed by all triggers. Additional module-specific parameters passed by module-specific triggers are described in the chapter for the specific module.
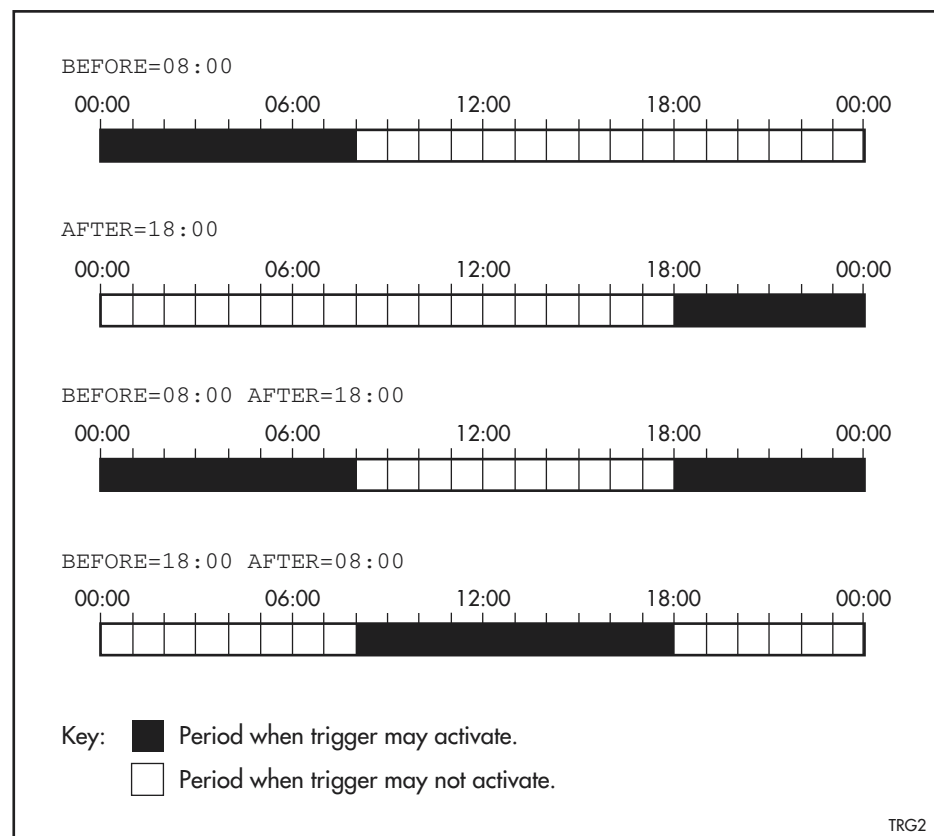
The **trigger** parameter specifies the number of the trigger to create. The number references the trigger in other commands. The specified trigger must not already exist. The **trigger** parameter must immediately follow the **create** keyword, and must be followed immediately by either **module**, **cpu**, **firewall**, **interface**, **memory**, **periodic**, **reboot**, or **time**.

The **after** parameter is a general trigger parameter that can be used with all trigger types except for **time** triggers. It specifies the earliest time of day in hours and minutes that the trigger is activated. The trigger may be activated any time between the time specified and midnight.

The **before** parameter is a general trigger parameter that can be used with all trigger types except for **time** triggers. It specifies the latest time of day in hours and minutes that the trigger is activated. The trigger may be activated any time between midnight and the time specified.

If neither **after** nor **before** are specified, there is no restriction on when the trigger may activate the script. If both **after** and **before** are specified, and the time periods specified overlap, the trigger may activate it any time during the overlap period (Figure 59-2).

Figure 59-2: The effects of different combinations of the **after** and **before** parameters in the **create trigger** and **set trigger** commands

The **circuit** parameter specifies a MIOX circuit to monitor. The trigger is activated when the condition defined by the combination of **interface**, **circuit**, and **event** occurs. The **circuit** parameter may only follow an **interface** parameter that specifies an X25T interface. When activated, the trigger passes three parameters to the trigger scripts—the X.25 instance that caused the trigger, the name of the MIOX circuit, and the event state.

The **cp** parameter specifies a PPP control protocol to monitor. The trigger is activated when the condition defined by the combination of **interface**, **cp**, and **event** occurs. The **cp** parameter may only follow an **interface** parameter that specifies a PPP interface. When activated, the trigger passes three parameters to the trigger scripts—the PPP instance that caused the trigger, the control protocol, and the event state.

The **cpu** parameter defines a CPU utilisation trigger and specifies the CPU utilisation level at which the trigger is to be activated. The **direction** parameter, and the general trigger parameters may also be specified.

The **date** parameter is a general trigger parameter that can be used with all trigger types. It specifies a date on which the trigger may activate. The **days** and **date** parameters are mutually exclusive—use one or the other.

The **days** parameter is a general trigger parameter that can be used with all trigger types. It specifies a comma-separated list of days when the trigger is activated. The **weekday** option is a synonym for the list **mon**, **tue**, **wed**, **thu**, **fri**; **weekend** is a synonym for the list **sat**, **sun**. Any combination of these and names of days is acceptable. The default is **all**. The **days** and **date** parameters are mutually exclusive—use one or the other.

The **direction** parameter specifies how the CPU or memory utilisation threshold is reached to activate the trigger. If **up** is specified, the trigger is activated when CPU or memory utilisation increases to or exceeds the threshold. If **down** is specified, the trigger is activated when CPU or memory utilisation decreases to or falls below the threshold. If **any** is specified, the trigger is activated when CPU or memory utilisation equals or passes the threshold in either direction. The default is **any**.

The **dlci** parameter specifies a Frame Relay DLC (Data Link Connection) to monitor. The trigger is activated when the condition defined by the combination of **interface**, **dlci**, and **event** occurs. The **dlci** parameter may only follow an **interface** parameter that specifies a Frame Relay interface. When activated, the trigger passes three parameters to the trigger scripts—the Frame Relay instance that caused the trigger, the DLCI, and the event state.

The **event** parameter is valid if the **interface** parameter or the **module** parameter is specified. The **event** parameter has different meanings in these two contexts. The **event** parameter is required after the **module** parameter, and in this context specifies the event for a module specific trigger. Values for the **event** parameter are described in the chapter for the given module. The **event** parameter is also required after the **interface** parameter, and in this context it specifies a link (interface) status change event. If **up** is specified, the trigger is activated when the interface becomes operational. If **down**, it is activated when the interface closes. If **fail** is specified, the trigger is activated when the interface fails to open at all for any reason. If **any** is specified, it is activated when any of these events occurs. The only valid events are **up** and **down** for the following interfaces:

■ Synchronous port (SYN*n*)

■ Basic Rate ISDN (BRI*n*)

■ Primary Rate ISDN (PRI*n*)

The **firewall** parameter defines a firewall trigger and specifies firewall events to monitor. The **firewall** parameter must be followed by the **mode** parameter. The trigger is activated when the condition defined by the combination of the **firewall** and **mode** parameters occurs. The general trigger parameters may

also be specified. When activated, the trigger passes two parameters to trigger scripts—the name of the firewall policy and the source IP address of the attack. The following table describes firewall events that are recognised.

Firewall Event	Description
DOSATTACK	Denial of service attack in which a remote user continually sends unwanted traffic.
FRAGATTACK	Attack using TCP fragments that are either too large or can never be reassembled.
HOSTSCAN	Scan of the hosts of the private network.
PORTSCAN	Portscan of the firewall or private network.
SESSION	This trigger activates when the first TCP session is created, and/or when the last active TCP session is closed. When all TCP sessions are closed, the router closes the link via which the TCP sessions were being transported. If the **mode** parameter is set to **start**, the trigger activates when the first TCP session is created. If **mode** is **end**, the trigger activates when the last TCP session is closed. This avoids the cost of unused dial-up links. **Caution** This trigger monitors TCP sessions only. The firewall may still be allowing other types of traffic.
SIPAUTOMAX	This trigger activates when the SIP ALG reaches the limit for the number of SIP clients it can support in automatic mode. After this trigger is first activated, further triggers are rate limited to once every 20 minutes. The trigger will not activate again until at least 20 minutes have passed in which the limit is not exceeded. Note that the firewall policy and source IP address script parameters are not valid for this type of event. You can set the **mode** parameter only to **start** for this trigger.
SMTPATTACK	Attack where email is received that is unwanted either because it is from a source identified as a source of spam, it is attempting to use a mail server as a third party relay, or it has a broadcast reply address.
SMURFATTACK	Directed attack on the hosts on the private network hidden by NAT.
SYNATTACK	Attack on a host using multiple opening TCP SYN packets to exhaust a host's available sessions or memory.
TCPATTACK	Attack on a host using TCP tiny fragments.

The **interface** parameter defines an interface (link) trigger and specifies the interface to monitor. The **event** parameter is required for an **interface** trigger. The **interface** parameter must be followed by the **event** parameter. The **cp** parameter may be used if **interface** specifies a PPP interface. The **circuit** parameter may be used if **interface** specifies an X.25T interface. The **dlci** parameter may be used if **interface** specifies a Frame Relay interface. The general trigger parameters may also be specified. Valid interfaces are:

■ BRI (such as bri0)

■ eth (such as eth0)

■ FR (such as fr0)

■ PPP (such as ppp0)

■ PRI (such as pri0)

■ syn (such as syn0)

■ VLAN (such as vlan1)

The interface must already exist. To see a list of all currently available interfaces, use the **show interface** command on page 9-72 of Chapter 9, Interfaces.

The **memory** parameter defines a memory utilisation trigger and specifies the percentage of free memory at which the trigger activates. The **direction** parameter and general trigger parameters may also be specified. Use the **show buffer** command on page 4-34 of Chapter 4, Configuring and Monitoring the System to display the current percentage of free memory.

The **mode** parameter specifies a firewall status change event and is valid following the **firewall** parameter. If **start** is specified, the trigger is activated when the event begins to occur since the last time period measured. If **end** is specified, the trigger is activated when the event stops after the last time period measured. If **both** is specified, it is activated when the firewall event begins or ends. The default is **both**. When the firewall trigger set is **sipautomax**, the only valid **mode** option is **start**.

The **module** parameter defines a module-specific trigger and specifies the name of the module to which this trigger applies. The module should be a valid router module as given in "Module identifiers, display names, and descriptions" on page B-2 of Appendix B, Reference Tables, and must be one for which module-specific triggers are defined. The **event** parameter must be specified, and other module parameters defining the module triggers may be valid or required for the given module. If the **module** parameter specified is **switch**, then the **port** parameter must be specified. The general trigger parameters may also be specified.

The **name** parameter is a general trigger parameter that can be used with all trigger types. It specifies a descriptive name for this trigger.

The **periodic** parameter defines a periodic trigger and specifies the period of the trigger in minutes. The general trigger parameters may also be specified.

The **port** parameter is valid if the **module** parameter is **switch**. The **port** parameter specifies the port where the event activates the trigger.

The **reboot** parameter defines a reboot trigger and specifies a list of reboot events that activates the trigger. If **crash** is specified, the trigger is activated by a router crash. If **restart** is specified, the trigger is activated by any reboot other than a router crash. If **all** is specified, the trigger is activated by any reboot event. The general trigger parameters may also be specified.

The **repeat** parameter is a general trigger parameter that can be used with all trigger types. It specifies whether the trigger repeats, and how many times the trigger repeats. If **yes** or **forever** is specified, the trigger repeats indefinitely, or until it is disabled. If **no** or **once** is specified, the trigger activates only once. If a numeric value is specified, the trigger repeats the set number of times. The default is **forever**. After the trigger has repeated for the specified number of times, the router disables the trigger so it no longer activates. If you create a configuration script using the **create config** command on page 5-22 of Chapter 5, Managing Configuration Files and Software Versions while the trigger is in this disabled state, the configuration script will create the trigger but set its **state** to **disabled**. Use the **enable trigger** command on page 59-17 to enable the trigger again.

The **script** parameter is a general trigger parameter that can be used with all trigger types. It specifies the name of a script to execute when the trigger is activated. A script is a predefined list of router commands. The specified script must already exist. The **script** parameter may be repeated up to five times in one command, to add up to five scripts (in the order specified) at once. Additional scripts may be added using the **add trigger** command on page 59-7.

The **state** parameter is a general trigger parameter that can be used with all trigger types. It specifies the initial state of the trigger. By default triggers are enabled when created. A trigger is activated automatically only when it is enabled. A trigger can be manually activated with the **activate trigger** command on page 59-6 regardless of whether the trigger is enabled or disabled.

The **test** parameter is a general trigger parameter that can be used with all trigger types. It specifies whether this trigger is in **test** mode. When in test mode, it activates and logs the trigger but does not execute configured scripts. The default is **no**.

The **time** parameter defines a time trigger and specifies the time of day in hours and minutes when the trigger is to be activated. Resolutions of up to one minute with an accuracy of five seconds are supported. The trigger is activated at most five seconds after the specified minute. The type parameters **date** or **days**, and the general parameters **script**, **name**, **repeat**, **state**, and **test** may also be specified. The **after** and **before** parameters are not valid, but all other general trigger parameters may also be specified.

Examples   To create trigger 1 that activates at 6am every weekday and initiates script offpeak.scp, use the command:

```
cre trig=1 ti=06:00 day=weekday sc=offpeak.scp rep=y
```

To create trigger 3 that executes script ipxcallg.scp when IPXCP closes on interface ppp3, use the command:

```
cre trig=3 int=ppp3 ev=down cp=ipxcp sc=ipxcallg.scp
```

To create trigger 6, which activates the script file fwsipmax.scp when the SIP ALG has reached the limit of SIP clients it is configured to support in automatic mode, use the command:

```
cre trig=6 fire=sipa mode=sta sc=fwsipmax.scp
```

To create module specific trigger 5 for the Q931 module, which executes script file spidok.scp when SPID initialisation occurs successfully on interface bri0, use the command:

```
cre trig=5 mod=Q931 ev=spidup int=bri0 sc=spidok.scp
```

Related Commands   **activate trigger**
add trigger
destroy trigger
disable trigger
enable trigger
set trigger
show trigger

delete trigger

Syntax   `DELete TRIGger=trigger-id NUMber=index`

where:

- *trigger-id* is a number from 1 to 250.

- *index* is a number from 1 to *n*, where *n* is the number of scripts assigned to the trigger.

Description   This command removes a script from a trigger. The **trigger** parameter specifies the number of the trigger from which the script is to be deleted. The specified trigger must already exist.

The **number** parameter specifies the position in the script list of the script to be removed. The specified script must already exist in that position.

Examples   To remove the third script from trigger 1, use the command:

    del trig=1 num=3

Related Commands   **add trigger**
destroy trigger
set trigger
show trigger

destroy trigger

Syntax   `DESTroy TRIGger=trigger-id`

where *trigger-id* is a number from 1 to 250.

Description   This command destroys a previously-defined trigger. The **trigger** parameter specifies the number of the trigger to destroy. The specified trigger must already exist.

Examples   To destroy trigger 1, use the command:

    dest trig=1

Related Commands   **add trigger**
create trigger
delete trigger
disable trigger
enable trigger
purge trigger
show trigger

disable trigger

Syntax DISable TRIGger[=*trigger-id*]

where *trigger-id* is a number from 1 to 250

Description This command disables the entire trigger facility if a trigger is not specified, or a specific trigger. The specified trigger will no longer be eligible for activation; however, it can still be manually activated with the **activate trigger** command.

The **trigger** parameter specifies the number of the trigger to disable. The specified trigger must already exist.

Examples To disable trigger 1, use the command:

```
dis trig=1
```

To disable the trigger module, use the command:

```
dis trig
```

Related Commands **activate trigger**
delete trigger
destroy trigger
enable trigger
purge trigger
show trigger

enable trigger

Syntax `ENAble TRIGger[=trigger-id]`

where *trigger-id* is a number from 1 to 250

Description This command enables the entire trigger facility, if a trigger is not specified, or a specific trigger.

NOTE: You must enable the trigger facility before you can enable any specific triggers.

When the trigger facility is enabled, you can specify an individual trigger. The specified trigger is then eligible for activation. All triggers are enabled by default when they are created. Except for manual activation with the **activate trigger** command, disabled triggers cannot be activated.

The **trigger** parameter specifies the number of the trigger to enable. The specified trigger must already exist.

Examples To enable the trigger module, use the command:

    ena trig

To enable trigger 1, use the command:

    ena trig=1

Related Commands **activate trigger**
delete trigger
destroy trigger
disable trigger
purge trigger
set trigger
show trigger

purge trigger

Syntax `PURge TRIGger`

Description This command erases the trigger facility configuration.

Related Commands **delete trigger**
destroy trigger
disable trigger
enable trigger
set trigger
show trigger

set trigger

Syntax

```
SET TRIGger=trigger-id [CPU[=value]] [DIrection={UP|DOWN|
    ANY}] [AFTer=hh:mm] [BEfore=hh:mm] [{DATe=date|
    DAYs=day-list}] [NAMe=name] [REPeat={Yes|No|ONCe|
    FORever|count}] [TEST={YES|NO|ON|OFF|TRUE|FALSE}]

SET TRIGger=trigger-id [FIREwall[={ALL|DOSattack|
    FRAgattack|HOStscan|PORtscan|SESSION|SIPAutomax|
    SMUrfattack|SYNattack|TCPattack}]] [MODE={STARt|END|
    BOTH}] [AFTer=hh:mm] [BEfore=hh:mm] [{DATe=date|
    DAYs=day-list}] [NAMe=name] [REPeat={Yes|No|ONCe|
    FORever|count}] [TEST={YES|NO|ON|OFF|True|False}]

SET TRIGGER=trigger-id [INTerface[=interface]] [EVent={UP|
    DOWN|FAIL|ANY}] [CIRCuit=miox-circuit] [CP={APPLE|ATCP|
    BCP|CCP|DCP|DNCP|IPcp|IPXcp|LCP}] [DLCi=dlci]
    [AFTer=hh:mm] [BEfore=hh:mm] [{DATe=date|
    DAYs=day-list}] [NAMe=name] [REPeat={Yes|No|ONCe|
    FORever|count}] [TEST={YES|NO|ON|OFF|True|False}]

SET TRIGger=trigger-id [MEMory[=value]] [DIrection={UP|
    DOWN|ANY}] [AFTer=hh:mm] [BEfore=hh:mm] [{DATe=date|
    DAYs=day-list}] [NAMe=name] [REPeat={Yes|No|ONCe|
    FORever|count}] [TEST={YES|NO|ON|OFF|True|False}]

SET TRIGger=trigger-id [MODule] [module-parameters...]
    [AFTer=hh:mm] [BEfore=hh:mm] [{DATe=date|
    DAYs=day-list}] [NAMe=name] [REPeat={Yes|No|ONCe|
    FORever|count}] [TEST={YES|NO|ON|OFF|True|False}]

SET TRIGger=trigger-id [PERiodic[=minutes]] [AFTer=hh:mm]
    [BEfore=hh:mm] [{DATe=date|DAYs=day-list}] [NAMe=name]
    [REPeat={Yes|No|ONCe|FORever|count}] [TEST={YES|NO|ON|
    OFF|True|False}]

SET TRIGger=trigger-id [REBoot[={REStart|CRASH|ALL}]]
    [AFTer=hh:mm] [BEfore=hh:mm] [{DATe=date|
    DAYs=day-list}] [NAMe=name] [REPeat={Yes|No|ONCe|
    FORever|count}] [TEST={YES|NO|ON|OFF|True|False}]

SET TRIGger=trigger-id [TIme[=hh:mm]] [{DATe=date|
    DAYs=day-list}] [NAMe=name] [REPeat={Yes|No|ONCe|
    FORever|count}] [TEST={YES|NO|ON|OFF|True|False}]
```

where:

- *trigger-id* is a number from 1 to 250.

- *module* is the name of a router module as given in "Module identifiers, display names, and descriptions" on page B-2 of Appendix B, Reference Tables.

- *event* is an event defined in the specified module

- *module-parameters* are one or more other required or optional parameters for the specified module and event, whose syntax is defined in the specified module.

- *value* is a number from 1 to 100.

- *interface* is a valid interface name.

- *miox-circuit* is an alphanumeric string 1 to 15 characters long.

- *dlci* is a Frame Relay Data Link Connection Identifier (DLCI) from 0 to 1023.

- *minutes* is a number from 1 to 1439.

- *hh:mm* is a time in hours and minutes.

- *date* is a date in the format dd-mmm-yyyy, where *mmm* is the first three letters of the month name.

- *day-list* is one or more keywords for the day of the week.

- *name* is a string 1 to 40 characters long. If the string contains spaces, it must be in double quotes.

- *count* is a number from 1 to 4 294 967 294 (2^{32}-2).

Description   This command modifies the definition of a trigger. Different trigger types are supported—CPU triggers, firewall triggers, link triggers, memory triggers, module triggers, periodic triggers, reboot triggers, and time triggers. Some of these have additional trigger-specific parameters. Parameters for module triggers are described in the chapter for the module that supports them. General trigger parameters can be specified for all trigger types: **after**, **before**, **days** or **date**, **name**, **repeat**, and **test**. The type of trigger cannot be changed.

The **trigger** parameter specifies the number of the trigger to create. The number references the trigger in other commands. The specified trigger must not already exist. The **trigger** parameter must immediately follow the **create** parameter.

The **after** parameter is a general trigger parameter that can be used with all trigger types except for **time** triggers. It specifies the earliest time of day in hours and minutes that the trigger is activated. The trigger is activated any time between the time specified and midnight.

The **before** parameter is a general trigger parameter that can be used with all trigger types except for **time** triggers. It specifies the latest time of day in hours and minutes that the trigger is activated. The trigger is activated any time between midnight and the time specified.

If neither **after** nor **before** are specified, there is no restriction on when the trigger may activate the script. If both **after** and **before** are specified, and the time periods specified overlap, the trigger may activate it any time during the overlap period (Figure 59-2 on page 59-10).

The **circuit** parameter specifies a MIOX circuit to monitor. The trigger is activated when the condition defined by the combination of **interface**, **circuit** and **event** occurs. The **circuit** parameter may only follow an **interface** parameter that specifies an X25T interface. When activated, the trigger passes three parameters to the trigger scripts—the X.25 instance that caused the trigger, the name of the MIOX circuit, and the event state.

The **cp** parameter specifies a PPP control protocol to monitor. The trigger activate when the condition defined by the combination of **interface**, **cp**, and **event** occurs. The **cp** parameter may only follow an **interface** parameter that specifies a PPP interface. When activated, the trigger passes three parameters to the trigger scripts—the PPP instance that caused the trigger, the control protocol, and the event state.

The **cpu** parameter defines a CPU utilisation trigger and specifies the CPU utilisation level at which the trigger is to be activated. The **direction** parameter and the general trigger parameters may also be specified.

The **date** parameter is a general trigger parameter that can be used with all trigger types. It specifies a date on which the trigger may activate. The **days** and **date** parameters are mutually exclusive—use one or the other.

The **days** parameter is a general trigger parameter that can be used with all trigger types. It specifies a comma-separated list of days when the trigger is activated. The **weekday** option is a synonym for the list **mon**, **tue**, **wed**, **thu**, **fri**; **weekend** is a synonym for the list **sat**, **sun**. Any combination of these and the day names is acceptable. The default is **all**. The **days** and **date** parameters are mutually exclusive—use one or the other.

The **direction** parameter specifies how the CPU or memory utilisation threshold is reached to activate the trigger. If **up** is specified, the trigger is activated when CPU or memory utilisation increases to or exceeds the threshold. If **down** is specified, the trigger is activated when CPU or memory utilisation decreases to or falls below the threshold. If **any** is specified, the trigger is activated when CPU or memory utilisation equals or passes the threshold in either direction. The default is **any**.

The **dlci** parameter specifies a Frame Relay DLC (Data Link Connection) to monitor. The trigger is activated when the condition defined by the combination of **interface**, **dlci**, and **event** occurs. The **dlci** parameter may only follow an **interface** parameter that specifies a Frame Relay interface. When activated, the trigger passes three parameters to the trigger scripts—the Frame Relay instance that caused the trigger, the DLCI, and the event state.

The **event** parameter after the **interface** parameter specifies a link (interface) status change event. If **up** is specified, the trigger is activated when the interface is operational. If **down**, it is activated when the interface closes. If **fail** is specified, the trigger is activated when the interface fails to open for any reason. If **any** is specified, it is activated when any of these events occur. The only valid events are **up** and **down** for the following interfaces:

■ Synchronous port (SYN*n*)

■ Basic Rate ISDN (BRI*n*)

■ Primary Rate ISDN (PRI*n*)

The **firewall** parameter defines a firewall trigger and specifies firewall events to monitor. The **firewall** parameter must be followed by the **mode** parameter. The trigger is activated when the condition defined by the combination of the **firewall** and **mode** parameters occurs. The general trigger parameters may also be specified. When activated, the trigger passes two parameters to the trigger scripts—the name of the firewall policy and the source IP address of the attack. The following table describes firewall events that are recognised.

Firewall Event	Description
DOSATTACK	Denial of service attack in which a remote user continually sends unwanted traffic.
FRAGATTACK	Attack using TCP fragments that are either too large or can never be reassembled.
HOSTSCAN	Scan of the hosts of the private network.
PORTSCAN	Portscan of the firewall or private network.

Firewall Event	Description
SESSION	This trigger activates when the first TCP session is created, and/or when the last active TCP session is closed. When all TCP sessions are closed, the router closes the link via which the TCP sessions were being transported. If the **mode** parameter is set to **start**, the trigger activates when the first TCP session is created. If **mode** is **end**, the trigger activates when the last TCP session is closed.This avoids the cost of unused dial-up links.
	Caution This trigger only monitors TCP sessions. The firewall may still be allowing other types of traffic.
SIPAUTOMAX	This trigger activates when the SIP ALG reaches the limit for the number of SIP clients it can support in automatic mode. After this trigger is first activated, further triggers are rate limited to once every 20 minutes. The trigger will not activate again until at least 20 minutes have passed in which the limit is not exceeded.
	Note that the firewall policy and source IP address script parameters are not valid for this type of event. You can set the **mode** parameter only to **start** for this trigger.
SMTPATTACK	Attack where email is received that is unwanted either because it is from a source identified as a source of spam, it is attempting to use a mail server as a third party relay, or it has a broadcast reply address.
SMURFATTACK	Directed attack on the hosts on the private network hidden by NAT.
SYNATTACK	Attack on a host using multiple opening TCP SYN packets to exhaust a host's available sessions or memory.
TCPATTACK	Attack on a host using TCP tiny fragments.

The **interface** parameter defines an interface (link) trigger and specifies the interface to monitor. The **event** parameter specifies a link (interface) status change event. The **circuit** parameter may be used if **interface** specifies an X.25 T interface. The **cp** parameter may be used if **interface** specifies a PPP interface. The **dlci** parameter may be used if **interface** specifies a Frame Relay interface. The general trigger parameters may also be specified. The type of trigger cannot be changed. Valid interfaces are:

■   BRI (such as bri0)

■   eth (such as eth0)

■   FR (such as fr0)

■   PPP (such as ppp0)

■   PRI (such as pri0)

■   syn (such as syn0)

■   VLAN (such as vlan1)

The **interface** parameter specifies a valid interface already assigned and configured. To see a list of current valid interfaces, use the **show interface** command on page 9-72 of Chapter 9, Interfaces.

The **memory** parameter defines a memory utilisation trigger and specifies the percentage of free memory at which the trigger activates. The **direction** parameter and general trigger parameters may also be specified. Use the **show buffer** command on page 4-34 of Chapter 4, Configuring and Monitoring the System to display the current percentage of free memory.

The **mode** parameter specifies a firewall status change event and is only valid following the **firewall** parameter. If **start** is specified, the trigger is activated when the event begins to occur since the last time period measured. If **end** is specified, the trigger is activated when the event stops after the last time period measured. If **both** is specified, the trigger is activated when the firewall event begins or ends. The default is **both**. When the firewall trigger set is **sipautomax**, the only valid option is **start**.

The **module** parameter specifies that the trigger to be modified is a module type trigger. The **event** parameter is not valid, but other module parameters may be specified. The general trigger parameters may also be specified.

The **name** parameter is a general trigger parameter that can be used with all trigger types. It specifies a descriptive name for this trigger.

The **periodic** parameter defines a periodic trigger and specifies the period of the trigger in minutes. The general trigger parameters may also be specified. The type of trigger cannot be changed.

The **port** parameter specifies the port where the event activates the trigger.

The **reboot** parameter defines a reboot trigger and specifies a list of reboot events that activates the trigger. If **crash** is specified, the trigger is activated by a router crash. If **restart** is specified, the trigger is activated by any reboot other than a router crash. If **all** is specified, the trigger is activated by any reboot event. The general trigger parameters may also be specified. The type of trigger cannot be changed.

The **repeat** parameter is a general trigger parameter that can be used with all trigger types. It specifies whether the trigger repeats, and how many times the trigger repeats. If **yes** or **forever** is specified, the trigger repeats indefinitely, or until it is disabled. If **no** or **once** is specified, the trigger activates only once. If a numeric value is specified, the trigger repeats the set number of times. The default is **forever**. After the trigger has repeated for the specified number of times, the router disables the trigger so it no longer activates. If you create a configuration script using the **create config** command on page 5-22 of Chapter 5, Managing Configuration Files and Software Versions while the trigger is in this disabled state, the configuration script will create the trigger but set its **state** to **disabled**. Use the **enable trigger** command on page 59-17 to enable the trigger again.

The **test** parameter is a general trigger parameter that can be used with all trigger types. It specifies whether this trigger is in **test** mode. When in test mode, it activates and logs the trigger but does not execute configured scripts. The default is **no**.

The **time** parameter specifies the time of day in hours and minutes when the time trigger is to be activated. Resolutions of up to one minute with an accuracy of five seconds are supported. The trigger is activated at most five seconds after the specified minute. The **after** and **before** parameters are not valid but other general trigger parameters can be specified. The type of trigger cannot be changed.

Examples   To modify time trigger 1 to activate at 8am every weekday, use the command:

```
set trig=1 ti=08:00 day=weekday rep=y
```

To set trigger 6 to only activate if the limit for the number of SIP clients that can be supported by the SIP ALG in automatic mode is reached between 8am and 5pm, use the command:

```
set trig=6 after=8:00 before=17:00
```

Related Commands   activate trigger
add trigger
create trigger
destroy trigger
disable trigger
enable trigger
show trigger

show trigger

Syntax   SHOW TRIGGER[=*trigger-id*] [{COUnter|FULl|STAtus|SUMmary}]

where *trigger-id* is a number from 1 to 250

Description   This command displays information about all triggers that have been configured, a specific one, or general configuration information about the trigger facility. The **trigger** parameter specifies the number of the trigger to display.

If neither trigger nor parameter is specified, or the **summary** parameter is specified, summary information for all or the specified triggers is displayed (Figure 59-3, Table 59-1). If **full** is specified, or a trigger is specified without the **summary** parameter, detailed information about the specified triggers is displayed (Figure 59-4 on page 59-25, Table 59-2 on page 59-25).

The **status** parameter displays general configuration information for the trigger facility (Figure 59-5 on page 59-26, Table 59-3 on page 59-26). A trigger identifier may not be specified.

The **counter** parameter displays counters for the trigger facility (Figure 59-6 on page 59-27, Table 59-4 on page 59-27). A trigger identifier may not be specified.

Figure 59-3: Example output from the **show trigger** command

```
TR# Type & Details              Name            En Te Rept #Scr Days/Date
--------------------------------------------------------------------------
001 Periodic (3 min)            Test Trigger    Y  Y  Yes   01  -TW-FSS
002 Time (10:00)                Call home       Y  N  Yes   01  23-Apr-2000
003 Reboot (Crash)              Get Debug Info  Y  N  Yes   01  MTWTFSS
004 Module (VRRP)               DOWNMASTER      Y  N  No    01  MTWTFSS
--------------------------------------------------------------------------
```

Table 59-1: Parameters in output of the **show trigger** command

Parameter	Meaning
TR#	Trigger identifier (ID).
Type & Details	Trigger type and details:
	Time - trigger time
	Periodic - period
	Reboot - either Crash, Reboot, or All
	Memory
	CPU
	Module - module, event
Name	Descriptive name of the trigger.
En	Whether the trigger is enabled.
Te	Whether the trigger is in test mode.
Rept	Whether the trigger repeats or a repeat count.The repeat count is decremented each time the trigger is activated automatically.
#Scr	Number of scripts associated with the trigger.
Days/Date	Days or date when the trigger is activated. For the days options, the days are shown as a seven character string representing Monday to Sunday. A hyphen indicates days when the trigger is not activated.

Figure 59-4: Example output from the **show trigger full** command

```
Trigger Configuration Details
-------------------------------------------------------------
Trigger ..................... 1
Name ....................... Bring up Wellington link
Type and details ........... Time (13:45)
Days ....................... All
Enabled .................... Enabled
Test ....................... No
Repeat ..................... No
Created/Modified ........... 08-Nov-1996 12:04:33
Number of Activations ....... 1
Last Activation ............. 08-Nov-1996 13:45:07
Number of scripts ........... 2

    callwgtn.scp
    idlewgtn.scp

Trigger ..................... 4
Name ....................... VRRP master down action
Type and details ........... Module (VRRP) DOWNMASTER
Other Parameters ........... VRID=1
Days ....................... Daily
Active TCP sessions ........ 0
Enabled .................... Enabled
Test ....................... No
Repeat ..................... No
Created/Modified ........... 25-Aug-2001 16:37:22
Number of Activations ....... 1
Last Activation ............. 08-Nov-1996 14:40:09
Number of scripts ........... 1

    downmast.scp
```

Table 59-2: Parameters in output of the **show trigger full** command

Parameter	Meaning
Trigger	Trigger identifier (ID).
Name	Descriptive name of the trigger.
Type and details	Trigger type and details
	Time - trigger time Periodic - period Reboot - either Crash, Reboot, or All Memory CPU Module - for module-specific triggers, the module name and the event on which the trigger is activated are included
Other parameters	Parameters for module-specific triggers, one per line, that define the trigger.
Days	List of the days when the trigger is activated, or either "Weekdays" (Monday to Friday), "Weekends" (Saturday and Sunday) or "Daily" (every day). Either "Days" or "Date" is displayed.
Date	Date or days when the trigger is activated.
Active TCP sessions	The total number of currently active TCP sessions on the router.

Table 59-2: Parameters in output of the **show trigger full** command (Continued)

Parameter	Meaning
Enabled	Whether the trigger is enabled.
Test	Whether the trigger is in test mode.
Repeat	Whether the trigger repeats.
Created/Modified	Date and time the trigger was created or last modified.
Number of Activations	Number of times the trigger has been activated (triggered) since the last router restart.
Last Activation	Date and time the trigger was last activated (triggered).
Number of scripts	Number of scripts assigned to the trigger, followed by a list of the script file names.

Figure 59-5: Example output from the **show trigger status** command

```
Trigger Module Configuration
------------------------------------------------------------
General
  Trigger Module .......... Enabled
  Triggers configured ...... 4
  Queued Commands .......... 0

Time Triggers
  Configured .............. 2
  Active .................. 2
  Activated today ......... 1

Periodic Triggers
  Configured .............. 1
  Active .................. 1
  Activated today ......... 0

Reboot Triggers
  Configured .............. 0

Interface Triggers
  Configured .............. 0

Resource Triggers
  Configured .............. 1
  Active .................. 1
  Activated today ......... 0

Module Triggers
   Configured .............. 2
   Activated today ......... 3
```

Table 59-3: Parameters in output of the **show trigger status** command

Parameter	Meaning
General	General information about the Trigger Facility.
Trigger Module	Whether the trigger module is enabled.
Triggers configured	Total number of triggers that have been configured.
Queued commands	Number of commands that are queued for execution.

Table 59-3: Parameters in output of the **show trigger status** command (Continued)

Parameter	Meaning
Time Triggers	Information about time triggers.
Periodic Triggers	Information about periodic triggers.
Reboot Triggers	Information about reboot triggers.
Interface Triggers	Information about interface triggers.
Resource Triggers	Information about CPU and memory resource triggers.
Module Triggers	Information about firewall and module triggers.
Configured	Number of triggers of the associated type that have been configured.
Active	Number of triggers of the associated type that are currently active (enabled).
Activated today	Number of times a trigger of the associated type has been activated (triggered) today.

Figure 59-6: Example output from the **show trigger counter** command

```
Trigger Module Counters
-----------------------------------------
Polls (05 sec timer) ............... 37
 Idle loop entry count .............. 5
 Time trigger checks ................ 2
 Time trigger queue rebuilds ........ 1
 Trigger activations ................ 1
 Time triggers activated today ...... 1
 Periodic triggers activated today .. 0
 Interface triggers activated today . 0
 Resource triggers activated today .. 0
 Module triggers activated today .... 3
```

Table 59-4: Parameters in output of the **show trigger counter** command

Parameter	Meaning
Polls (05 sec timer)	Number of times the trigger module has polled for a trigger activation event.
Idle loop entry count	Number of times the trigger module has prepared commands for execution.
Time trigger checks	Number of times the trigger module has checked the list of time triggers for a trigger to activate.
Time trigger queue rebuilds	Number of times the time trigger queue has been rebuilt because time triggers have been added, deleted or modified, or because the time/date has been changed.
Trigger activations	Number of times a trigger has been activated.
Time triggers activated today	Number of times a time trigger has been activated today.
Periodic triggers activated today	Number of times a periodic trigger has been activated today.
Interface triggers activated today	Number of times an interface trigger has been activated today.

Table 59-4: Parameters in output of the **show trigger counter** command (Continued)

Parameter	Meaning
Resource triggers activated today	Number of times a CPU or memory resource trigger has been activated today.
Module triggers activated today	Number of times a firewall or module trigger has been activated today.

Examples   To display summary information for trigger 3, use the command:

```
sh trig=3 sum
```

To display summary information for all triggers, use the command:

```
sh trig
```

To display a detailed description of trigger 3, use the command:

```
sh trig=3
```

To display a detailed description of all triggers, use the command:

```
sh trig ful
```

To display general configuration information for the trigger facility, use the command:

```
sh trig sta
```

To display counters for the trigger facility, use the command:

```
sh trig cou
```

Related Commands   **activate trigger**
add trigger
create trigger
delete trigger
destroy trigger
disable trigger
enable trigger
purge trigger
set trigger