

Triggers

Feature Overview and Configuration Guide

Introduction

This guide provides information about the Trigger facility on AlliedWare Plus™ devices.

The Trigger facility provides a powerful mechanism for automatic and timed management of your device by automating the execution of commands in response to certain events. For example, you can use triggers to deactivate a service during the weekends, or to collect diagnostic information when the CPU usage is high.

A **trigger** is an ordered sequence of scripts that is executed when a certain event occurs. A **script** is a sequence of commands stored as a plain text file on a file subsystem accessible to the device, such as Flash memory. Each trigger may reference multiple scripts and any script may be used by any trigger. When an event activates a trigger, the trigger executes the scripts associated with it in sequence. One script is executed completely before the next script begins. Various types of triggers are supported, each activated in a different way.

Content

Introduction	1
Products and software version that apply to this guide	2
Configuring a Trigger	3
General steps	3
Environment sensor triggers	6
Triggers for events related to PoE Power Sourcing Equipment.....	8
Regular expressions in log type triggers.....	9
Troubleshooting Triggers	9
Triggers Configuration Examples	11
Restrict Internet Access	11
Capture Unusual CPU and RAM Activity	12
See Daily Statistics.....	14
Turn off Power to Port LEDs.....	15

Products and software version that apply to this guide

This guide applies to all AlliedWare Plus™ products, running version 5.4.4 or later.

For more information, see the following documents:

- The [product's Datasheet](#)
- The product's [Command Reference](#)

These documents are available from the above links on our website at alliedtelesis.com.

Most features described in this document are supported from AlliedWare Plus 5.4.4 or later. These features are available in later releases:

- Versions 5.4.7-2.1 and later support log message triggers.
- Versions 5.4.8-2.1 and later support day of the month triggers.
- Versions 5.5.1-1.1 and later support guest node triggers.
- Versions 5.5.2-1.1 and later support environmental sensor triggers.
- Versions 5.5.3-1.1 and later support PoE PSE triggers.

Configuring a Trigger

General steps

The following describes the **general** steps to configure a trigger. For specific configuration examples, see ["Triggers Configuration Examples" on page 11](#).

Step 1: Create a command script

- Create a command script with the commands you would like executed when the trigger conditions are met. To create the command script using the CLI, use the command:

```
awplus# edit [<filename>]
```

- Alternatively, you can create a script on a PC then load it onto your device using the **copy (URL)** command.

Step 2: Enter the trigger configuration mode

- You must be in Global Configuration mode to reach Trigger Configuration mode; use the command:

```
awplus# configure terminal
```

- To create a trigger and enter its configuration mode, use the command:

```
awplus(config)# trigger <1-250>
```

Step 3: Set the trigger type

The trigger type determines how the trigger is activated. To set the trigger to activate:

- when a Secure Digital (SD) or Secure Digital High Capacity (SDHC) card is either inserted or removed, use the command:

```
awplus(config-trigger)# type card {in|out}
```

- when CPU usage reaches a certain level, use the command:

```
awplus(config-trigger)# type cpu <1-100> [up|down|any]
```

- when the link status of a particular interface changes, use the command:

```
awplus(config-trigger)# type interface <interface> [up|down|any]
```

- when the RAM usage reaches a certain level, use the command:

```
awplus(config-trigger)# type memory <1-100> [up|down|any]
```

- periodically after a set number of minutes, use the command:

```
awplus(config-trigger)# type periodic <1-1440>
```

- when a ping poll identifies that connectivity to a target device has changed, use the command:

```
awplus(config-trigger)# type ping-poll <1-100> {up|down}
```

- if your device reboots, use the command:

```
awplus(config-trigger)# type reboot
```

- if an environment sensor changes state, use the command:

```
awplus(config-trigger)# type env-sensor [node <number>]
resource <resource-id> sensor <sensor-id> state {true|false|any}
```

See ["Environment sensor triggers" on page 6](#).

- if an environment sensor crosses a threshold, use the command:

```
awplus(config-trigger)# type env-sensor [node <number>]
resource <resource-id> sensor <sensor-id> {low-limit|high-limit}
{exceeded|cleared|any}
```

See ["Environment sensor triggers" on page 6](#).

- on a PoE switch, if the total amount of power being drawn exceeds the switch's power budget, use the command:

```
awplus(config-trigger)# type main-pse {up|down|any}
```

See ["Triggers for events related to PoE Power Sourcing Equipment" on page 8](#).

- on a PoE switch, if the amount of power being drawn by a port exceeds the capabilities of the port, use the command:

```
awplus(config-trigger)# type pse-port <port-number> {up|down|any}
```

See ["Triggers for events related to PoE Power Sourcing Equipment" on page 8](#).

- when a stacking link goes up or down, use the command:

```
awplus(config-trigger)# type stack link {up|down}
```

- when a device joins or leaves the stack, use the command:

```
awplus(config-trigger)# type stack member {join|leave}
```

- when the device enters the fail-over state, use the command:

```
awplus(config-trigger)# type stack master-fail
```

- when a stack member becomes a disabled master, use the command:

```
awplus(config-trigger)# type stack disabled-master
```

- when a Control Fabric Card (CFC) or Line Interface Card (LIF) is inserted or removed from a chassis, use the command:

```
awplus(config-trigger)# type chassis card {join|leave}
```

- when the active Control Fabric Card (CFC) in a chassis fails, use the command:

```
awplus(config-trigger)# type chassis active-CFC-fail
```

- when a CFC in the chassis becomes a disabled CFC, use the command:

```
awplus(config-trigger)# type chassis disabled-CFC
```

- when an AMF node join or leave event occurs in a working set, use the command:

```
awplus(config-trigger)# type atmf node {join|leave}
```

Caution: Only configure an AMF node trigger on one device because it is a network wide event.

- when an AMF guest node join or leave event occurs, use the command:

```
awplus(config-trigger)# type atmf guest {join|leave}
```

- if USB storage device is inserted or removed, use the command:

```
awplus(config-trigger)# type usb {in|out}
```

- if a particular string is generated in a log message of severity level notice (5) or higher, use the command:

```
awplus(config-trigger)# type log <log-message-string>
```

Regular expressions (PCRE) are fully supported in the log message string ("[Regular expressions in log type triggers](#)" on page 9).

Log type triggers are limited to activating at most once per second.

- when ISSU has upgraded the CFCs, has completed the upgrade of all cards, or has aborted the upgrade, use the command:

```
awplus(config-trigger)# type issu {cfcs-upgraded|completed|aborted}
```

Caution: ISSU trigger scripts that use any type other than "issu completed" must not place the device into Configuration Mode.

To see more information about the ISSU upgrade process, or reasons why the ISSU may abort, see <http://www.alliedtelesis.com/documents/how-use-service-software-upgrade-issu-feature>.

- at a specific time of the day, use the command:

```
awplus(config-trigger)# type time <hh:mm>
```

Note that a combined limit of 10 triggers of the type periodic and type time can be configured. If you attempt to add more than 10 triggers the following error message is displayed:

```
% Cannot configure more than 10 triggers with the type time or periodic
```

Step 4: Set the time and days that the trigger can activate on

- By default triggers can activate at any time of the day, on all days. If you want your trigger to activate only during a specific time of the day, use the command:

```
awplus(config-trigger)# time {[after <hh:mm:ss>]
[before <hh:mm:ss>]}
```

- If you want your trigger to activate only on a specific date, use the command:

```
awplus(config-trigger)# day <1-31> <month> <2000-2035>
```

- If you want the trigger to activate only on specific days of the week, use the command:

```
awplus(config-trigger)# day <weekday>
```

- If you want the trigger to activate only on a specific day each month, use the command:

```
awplus(config-trigger)# day <1-31>
```

- If you want the trigger to activate only on a specific day and month each year, use the command:

```
awplus(config-trigger)# day <1-31> <month>
```

Note that the **day** command can only be used once with each trigger.

Step 5: Specify how often the trigger can activate

- By default, triggers can activate an unlimited number of times, as long as the trigger conditions are met. To set a limit on the number of times a trigger can activate, use the command:

```
awplus(config-trigger)# repeat {forever|no|once|yes|<1-4294967294>}
```

- Your device maintains two counters that track the number of times a trigger has activated. One counts the total number of times the trigger is activated and is only reset if the device restarts, or when the trigger is destroyed. The other counter tracks the permitted number of repetitions. To reset this counter, use the **repeat** command.

Step 6: Add the scripts to the trigger

- You can add up to five command scripts to the trigger. When a trigger is activated, it executes the scripts in sequence, with the lowest numbered script activated first. The first script runs to completion before the next script begins. To add a script, use the command:

```
awplus(config-trigger)# script <1-5> <filename>
```

Script file names must end in '.scp' or '.sh'.

Note that a script activated from an SD card trigger cannot be located on the SD or SDHC card. It must be located in Flash memory.

Step 7: Specify a description for the trigger

- Specify a description for the trigger, so that you can easily identify the trigger in show commands and log output. Use the command:

```
awplus(config-trigger)# description <description>
```

Step 8: Verify the trigger's configuration

- To check the configuration of the trigger, use the command:

```
awplus(config-trigger)# show trigger [<1-250>|counter|full]
```

Environment sensor triggers

From 5.5.2-1.1 onwards, you can create triggers that will activate when the device's environment sensors detect an event, and run commands of your choice. Environment sensor events are shown in the output of the command **show system environment**, and include device temperature, power settings, voltage, and fan speed.

Depending on the device and sensor, you can create a trigger to run when:

- the sensor's state changes, for example when a loss of power is detected for a power supply, or when power is restored, or both.
- the sensor's reading crosses a high or low threshold, for example when the device temperature becomes too high, or returns to normal, or both.

Before creating the new trigger, see what sensors your device has and find the resource ID and sensor ID, using the command:

```
awplus# show system environment
```

For example, to create a trigger for low speed in fan 2 on an x930-52GTX switch, use resource ID 3 and sensor ID 2, as highlighted in the following output:

```
awplus# show system environment
...
Resource ID: 3 Name: AT-x930-52GTX
ID Sensor (Units) Reading Low Limit High Limit Status
1 Fan: SYS Fan 1 (Rpm) 5579 4141 - Ok
2 Fan: SYS Fan 2 (Rpm) 5533 4141 - Ok
3 Voltage: 1.5V (Volts) 1.484 1.354 1.654 Ok
4 Voltage: Battery (Volts) 3.094 2.700 3.586 Ok
```

State change triggers

The command to create a trigger when an environment sensor changes state is:

```
awplus(config-trigger)# type env-sensor [node <number>]
resource <resource-id> sensor <sensor-id> state {true|false|any}
```

This command lets you specify:

Parameter	Description
node	The CFC, line card or VCStack member ID that the sensor is on, for example, 1.5 for card 5 on unit 1, or 3 for stack member 3. You can leave this parameter out on standalone devices.
state true	The trigger will trigger if this sensor reading changes to 'Yes' or 'Open' in the output of show system environment
state false	The trigger will trigger if this sensor reading changes to 'No' or 'Closed' in the output of show system environment
any	The trigger will trigger if the sensor reading changes to any of the above.

Threshold change triggers

The command to create a trigger when an environmental sensor crosses a threshold is:

```
awplus(config-trigger)# type env-sensor [node <number>]
resource <resource-id> sensor <sensor-id> {low-limit|high-limit}
{exceeded|cleared|any}
```

This command lets you specify:

Parameter	Description
node	The CFC, line card or VCStack member ID that the sensor is on, for example, 1.5 for card 5 on unit 1, or 3 for stack member 3. You can leave this parameter out on standalone devices.
low-limit	The trigger will trigger when the sensor reading falls below the 'Low Limit' alarm threshold shown in show system environment , or returns to an acceptable value after being too low, or both. The alarm threshold values are pre-defined within the device and cannot be changed.
high-limit	The trigger will trigger when the sensor reading rises above the 'High Limit' alarm threshold shown in show system environment , or returns to an acceptable value after being too high, or both. The alarm threshold values are pre-defined within the device and cannot be changed.

Parameter	Description
exceeded	If you chose low-limit, the trigger will trigger if the sensor's reading falls below the low limit. If you chose high-limit, the trigger will trigger if the sensor's reading goes above the high limit.
cleared	If you chose low-limit, the trigger will trigger if the sensor's reading rises to above the low limit again. If you chose high-limit, the trigger will trigger if the sensor's reading falls below the high limit again. Some temperature sensors include a hysteresis value and will not clear until the temperature has changed significantly. For example, if a sensor has a high alarm threshold of 75 degrees Celsius, the hysteresis value may mean that the alarm clears when the temperature falls to 63 degrees Celsius.
any	The trigger will trigger if the low or high limit is either exceeded or cleared.

Triggers for events related to PoE Power Sourcing Equipment

From version 5.5.3-1.1 onwards, you can create triggers for PoE related events:

- **main-pse** - this monitors the switch's total PoE power budget. It activates if the total amount of power being drawn exceeds the switch's power budget.
- **pse-port** - this monitors individual ports. It activates if the amount of power being drawn by a port exceeds the capabilities of the port.

You can also activate these trigger types when the amount of power drops to within the limit again. Note that the switch will take corrective action when the power budget or port limit is exceeded, such as shutting down the affected port. This means the amount of power will drop soon after the fault occurs.

To create a trigger that monitors the total PoE power budget, use the command:

```
awplus(config-trigger)# type main-pse {up|down|any}
```

On switches that are in a VCStack, that trigger will activate if any stack member meets its conditions. You can monitor a specific stack member by using the command:

```
awplus(config-trigger)# type main-pse member <id> {up|down|any}
```

To create a trigger that monitors a particular port, use the command:

```
awplus(config-trigger)# type pse-port <port-number> {up|down|any}
```

In all of these commands:

- **up** means the trigger activates when the power becomes too high
- **down** means the trigger activates when the power drops low enough again
- **any** means it activates both times - when the power becomes too high and when it drops low enough again.

Regular expressions in log type triggers

Log type triggers fully support regular expressions using PCRE (Perl-Compatible Regular Expression) syntax. The following table shows some of the common syntax elements.

Syntax	Description
.	Use for any character.
\d	Use for any number.
*	Use for 0 or more occurrences of the preceding element.
+	Use for 1 or more occurrences of the preceding element.
()	Use parentheses for grouping.
	Use to separate alternatives (or).
[]	Use brackets to enclose a set of characters.

Examples Regular expressions can be used in the log message string to select a log messages to activate a script. The following examples show possible log type triggers using regular expressions.

To activate a trigger when a log message includes:

- a port identifier followed by a 'failed' message, use the command:

```
awplus(config-trigger)# type log port.+ failed
```

- a 'joined' message from any stack member:

```
awplus(config-trigger)# type log Stack member \d has joined
```

- a 'joined' message from a set of specific stack members (1, 2 or 3), use the command:

```
awplus(config-trigger)# type log Stack member [1-3] has joined
```

- a message mentioning an interface including either 'failed' or 'succeeded', use the command:

```
awplus(config-trigger)# type log Interface [a-z]* {succeeded|failed}
```

Troubleshooting Triggers

You can use the trigger diagnostic mode and trigger debugging to test your triggers and troubleshoot any issues.

Diagnostic mode is set per trigger. In this mode the trigger activates if its trigger conditions are met, but does not run any of its scripts. Your device generates a log message to indicate that the trigger was activated.

- To place a trigger in diagnostic mode, enter the trigger's configuration mode and use the command:

```
awplus(config-trigger)# test
```

- To start debugging for triggers, use the command:

```
awplus(config-trigger)# debug trigger
```

- This generates detailed messages about how your device is processing the trigger commands and activating the triggers.

Enabling and Disabling

Triggers are enabled by default. This allows the trigger to activate as soon as its trigger conditions are met. If you need to disable a trigger but do not want to delete the trigger, use the command:

```
awplus(config-trigger)# no active
```

- To enable the trigger again, use the command:

```
awplus(config-trigger)# active
```

- To delete the trigger, use the command

```
awplus(config-trigger)# no trigger <1-250>
```

Triggers Configuration Examples

The section describes how to configure triggers to:

- "Restrict Internet Access" on page 11
- "Capture Unusual CPU and RAM Activity" on page 12
- "See Daily Statistics" on page 14
- "Turn off Power to Port LEDs" on page 15

Restrict Internet Access

In the following configuration the ACME company wants to restrict its employees from accessing popular video sharing websites as this is causing bandwidth problems during work hours. The ACME company is happy for workers to access the site after work hours.

Employee PCs at ACME are on vlan2. Two triggers with associated scripts are needed:

- Trigger 1 activates at 8.30am and runs a script called **shutdown.scp**. This script adds commands to restrict access to the specified sites.
- Trigger 2 activates at 5.30pm and runs the script called **open.scp**. This script removes the configuration specified by shutdown.scp.

1. Create the **shutdown.scp** script.

- Create a configuration script using Access Control List commands to restrict users on vlan2 from accessing the specific sites.

2. Create the **open.scp** script.

- Create a script to remove the ACL configuration specified in the **shutdown.scp** file.

3. Configure trigger 1.

- To create trigger 1, use the commands:

```
awplus# configure terminal
awplus(config)# trigger 1
```

- Set the trigger to activate at 8:30am, by using the command:

```
awplus(config-trigger)# type time 08:30
```

- Set the trigger to activate on Monday, Tuesday, Wednesday, Thursday and Friday:

```
awplus(config-trigger)# day mon tue wed thur fri
```

- Add the script **shutdown.scp** to the trigger:

```
awplus(config-trigger)# script 1 shutdown.scp
```

- Specify a helpful description, such as **Stops access to video sharing sites**. Use the command:

```
awplus(config-trigger)# description Stops access to video sharing sites
```

- Change to Global Configuration mode:

```
awplus(config-trigger)#exit
```

4. Configure trigger 2.

- To create trigger 2, use the command:

```
awplus(config)# trigger 2
```

- Set the trigger to activate at 5.30pm

```
awplus(config-trigger)# type time 17:30
```

- Set the trigger to activate on Monday, Tuesday, Wednesday, Thursday and Friday:

```
awplus(config-trigger)# day mon tue wed thur fri
```

- Add the script **open.scp** to the trigger:

```
awplus(config-trigger)# script 1 open.scp
```

- Specify a helpful description, such as **Access allowed to video sharing sites**. Use the command:

```
awplus(config-trigger)# description Access allowed to video sharing sites
```

5. Verify the configuration.

- To check the configuration of the triggers, use the commands:

```
awplus# show trigger 1
```

```
awplus# show trigger 2
```

Capture Unusual CPU and RAM Activity

The following configuration allows you to troubleshoot high CPU or RAM usage by the device. It uses two triggers to capture show output, and places this output in a file.

- Trigger 3 activates the script `cpu-usage.scp` when CPU usage is over 90% and can activate up to 5 times
- Trigger 4 activates the script `ram-usage.scp` when RAM usage is over 95%, and can activate up to 10 times

1. Create the `cpu-usage.scp` configuration script

- Create a script with the appropriate show command:

```
awplus# show cpu | redirect showcpu.txt
```

The output of the **show cpu** command has been redirected into a file. It is not possible to display trigger script output on the terminal. Redirecting the command output to a file means it is available for later inspection.

Note that the files may grow large accumulating data and consume available Flash memory.

2. Create the ram-usage.scp configuration script.

- Create a script with the appropriate show command.

```
awplus# show memory | redirect showmem.txt
```

The output of the **show memory** command has been redirected into a file. It is not possible to display trigger script output on the terminal. Redirecting the command output to a file means it is available for later inspection.

3. Configure trigger 3.

- To create trigger 3, use the commands.

```
awplus# configure terminal
awplus(config)# trigger 3
```

- Set the trigger to activate when CPU usage exceeds 80%

```
awplus(config-trigger)# type cpu 90 up
```

- Add the script **cpu-usage.scp** to the trigger:

```
awplus(config-trigger)# script 1 cpu-usage.scp
```

- Return to Global Configuration mode:

```
awplus(config-trigger)# exit
```

4. Configure trigger 4.

- To create trigger 4, use the command:

```
awplus(config)# trigger 4
```

- Set the trigger to activate when RAM usage exceeds 95%

```
awplus(config-trigger)# type cpu 95 up
```

- Add the script **cpu-usage.scp** to the trigger:

```
awplus(config-trigger)# script 1 ram-usage.scp
```

5. Verify the configuration.

- To check the configuration of the triggers, use the command:

```
awplus#show trigger 3
awplus#show trigger 4
```

See Daily Statistics

The ACME company has recently set up QoS on its traffic to give traffic different priorities to the ISP. ACME wants to assess how much traffic is dropped with the QoS bandwidths set over the next week. To do this, they want to generate an hourly report on QoS traffic on the first day that this is implemented.

- Trigger 5 activates the script **qos-stats.scp** every 60 minutes.

The trigger is set to only activate during work hours.

1. Create the **qos-stats.scp** script

- Create a configuration script with the appropriate show commands. You can either create the configuration script using the CLI with the **edit** command or create a script on a PC then load it onto your device using the **copy (URL)** command.

2. Configure trigger 5

- To create trigger 5, use the commands:

```
awplus(config)#configure terminal
awplus(config)#trigger
```

- Set the trigger to activate periodically every 60 minutes:

```
awplus(config-trigger)#type periodic 60
```

- Set the trigger to activate only during the hours of 8:00am and 6:00pm:

```
awplus(config-trigger)#time after 8:00 before 18:00
```

- Add the script **qos-stats.scp** to the trigger:

```
awplus(config-trigger)# script 1 qos-stats.scp
```

3. Verify the configuration:

- To check the configuration of the trigger, use the command:

```
awplus(config)#show trigger
```

Turn off Power to Port LEDs

The following configuration allows you to conserve power by using the eco-friendly LED (Light Emitting Diode) feature to turn off power to the port LEDs during non-work hours.

- Trigger 6 activates at 5.30pm and runs a script called **LEDOff.scp**. This script adds commands to turn off power to all the port LEDs
- Trigger 7 activates at 8.30am and runs the script called **LEDOn.scp**. This script removes the configuration specified by **LEDOff.scp**

1. Create the **LEDOff.scp** script

Create a configuration script with the commands that are executed when the trigger conditions are met. You can either create the configuration script using the CLI with the **edit** command or create a script on a PC then load it onto your device using the **copy (URL)** command. The configuration script for this example is:

```
!
enable
configure terminal
ecofriendly led
exit
exit
!
```

2. Create the **LEDOn.scp** script

Create a script to remove the configuration specified in the **LEDOff.scp** file. The configuration script for this example is:

```
!
enable
configure terminal
no ecofriendly led
exit
exit
!
```

3. Configure trigger 6

- To create trigger 6, use the commands:

```
awplus(config)#configure terminal
awplus(config)#trigger 6
```

- Set the trigger to activate at 5:30pm, by using the command

```
awplus(config-trigger)#type time 17:30
```

- Set the trigger to activate on Monday, Tuesday, Wednesday, Thursday and Friday:

```
awplus(config-trigger)#day mon tue wed thur fri
```

- Add the script **LEDOff.scp** to the trigger:

```
awplus(config-trigger)#script 1 LEDOff.scp
```

- Specify a helpful description, such as **Shutdown power to LEDs**. Use the command:

```
awplus(config-trigger)#description Shutdown power to LEDs
```

- Change to Global Configuration mode:

```
awplus(config-trigger)# exit
```

4. Configure trigger 7:

- To create trigger 7, use the command:

```
awplus(config)#trigger 9
```

- Set the trigger to activate at 8.30am:

```
awplus(config-trigger)#type time 08:30
```

- Set the trigger to activate on Monday, Tuesday, Wednesday, Thursday and Friday:

```
awplus(config-trigger)#day mon tue wed thur fri
```

- Add the script **LEDon.scp** to the trigger:

```
awplus(config-trigger)#script 1 LEDon.scp
```

- Specify a helpful description, such as **Turn on power to LEDs**. Use the command:

```
awplus(config-trigger)#description Turn on power to LEDs
```

5. Verify the configuration:

- To check the configuration of the triggers, use the commands:

```
awplus(config)#show trigger 6
```

```
awplus(config)#show trigger 7
```