Introduction

AlliedWare Plus™ has a comprehensive debugging and logging facility in various protocols and components. This guide describes how to start and stop debugging and logging.

Products and software version that apply to this guide

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

Feature support and implementation varies between products. To see whether a product supports a particular feature or command, 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:

- Log type triggers are available from 5.4.7-2.1 onwards. See “Activating Configuration Scripts with Log Message Triggers” on page 15
- External logging is available from 5.4.7-1.1 onwards. See “External logging”
- The commands copy buffered-log and copy permanent-log are available from 5.4.7-1.1 onwards
- The exclude filter option is available from 5.4.4-4.13 onwards. See “Configuring filters to exclude messages from different types of logging output” on page 13.
Content

Introduction .........................................................................................................................................1
  Products and software version that apply to this guide ...............................................................1
Debugging ...........................................................................................................................................3
  Displaying debug on a terminal ...................................................................................................3
  Turning off debugging ..............................................................................................................4
Logging ...............................................................................................................................................4
  Reading log messages ...............................................................................................................4
  Log outputs ...................................................................................................................................5
Filtering log message output .........................................................................................................12
  Configuring filters to include messages in different types of logging output .............................12
  Configuring filters to exclude messages from different types of logging output.......................13
  Restoring default settings ........................................................................................................14
Activating Configuration Scripts with Log Message Triggers ......................................................15
  Configuring a log message trigger ............................................................................................15
  Regular expressions in log type triggers ..................................................................................16
Debugging

Many protocols have debug commands which log protocol-specific information. For example, using the `debug mstp protocol` command results in the device writing all debugging messages generated by the MSTP algorithm to the logging system.

On using a debug command, the protocol continues to generate output until the `no` parameter is used with the command.

Displaying debug on a terminal

To display debug output on the terminal:

**Step 1: Turn on the debug options by using the relevant debug command**

```
awplus#debug <protocol> <parameter>
```

**Step 2: Run the terminal monitor command**

```
awplus#terminal monitor
```

**Sample Output**

This is a sample output of the `debug rsvp events` command displayed on the terminal:

```
awplus#terminal monitor

Dec 2 16:41:49 localhost RSVP[6518]: RSVP: RSVP message sent to 10.10.23.60/32 via interface vlan2

Dec 2 16:41:57 localhost RSVP[6518]: RSVP: Received an RSVP message of type RSVP Reservation from 192.168.0.60 via interface vlan2

Dec 2 16:41:57 localhost RSVP[6518]: RSVP: Received a RESV message from 10.10.23.60/32

```

The debug output will only come to the console while the terminal monitor mode is enabled.

Terminal monitor mode is disabled by the command:

```
awplus#terminal no monitor
```

Additionally, it is possible to enable terminal monitor mode for a specified number of seconds, after which it is automatically disabled. For example, to enable terminal monitor mode for just 30 seconds, use the command:

```
awplus#terminal monitor 30
```
Turning off debugging

To turn off debugging, use the `no debug` or `undebug` command. When a protocol is specified with the `no debug` or `undebug` commands, debugging is stopped for the specified protocol.

- For example, to turn off STP debug (for STP, RSTP and MSTP), use the command:
  ```
  awplus(config)#no debug mstp
  ```
- To turn off AMF debug, use the command:
  ```
  awplus(config)#no debug atmf
  ```
- To stop all debugging, use the `all` parameter with these commands.
  ```
  awplus#undebug all
  ```

Logging

Protocols generate important debugging messages by default, and send them to the logging system. Log messages can be filtered based on: the program that generated the message, the severity level of the message, the type of facility that generated the message, and substrings within the message text.

Reading log messages

Log messages generated by AlliedWare Plus show information in the following format:

```
<date> <time> <facility>..<severity> <hostname> <program>[:<pid>]: <message>
```

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;date&gt;</code></td>
<td>The date and time when the log message was generated, according to the device’s clock.</td>
</tr>
<tr>
<td><code>&lt;time&gt;</code></td>
<td>The facility assigned for the message.</td>
</tr>
<tr>
<td><code>&lt;facility&gt;</code></td>
<td>The severity level of the message, indicating its importance.</td>
</tr>
<tr>
<td><code>&lt;hostname&gt;</code></td>
<td>The device’s hostname, as configured by the <code>hostname</code> command (default: awplus).</td>
</tr>
<tr>
<td><code>&lt;program&gt;</code></td>
<td>Within the modular operating system, the particular program that generated the message. Some programs correspond to particular features (e.g., MSTP, EPSR), while others correspond to internal functions in the operating system (e.g. kernel).</td>
</tr>
<tr>
<td><code>&lt;pid&gt;</code></td>
<td>The process ID (PID) of the current instance of the software program that generated the message. A particular process ID does not always correspond to the same program. Some log messages, such as kernel messages, may not include a process ID.</td>
</tr>
<tr>
<td><code>&lt;message&gt;</code></td>
<td>The specific content of the log message. This may include some variable elements, such as interface names, and some strings that are fixed.</td>
</tr>
</tbody>
</table>
These are the severity levels for log messages:

<table>
<thead>
<tr>
<th>SEVERITY IN MESSAGE</th>
<th>SEVERITY LEVEL</th>
<th>MEANING</th>
</tr>
</thead>
<tbody>
<tr>
<td>emerg</td>
<td>0</td>
<td>Emergency: system is unusable; operation severely impaired.</td>
</tr>
<tr>
<td>alert</td>
<td>1</td>
<td>Alert: action must be taken immediately; operation has been or could be affected.</td>
</tr>
<tr>
<td>crit</td>
<td>2</td>
<td>Critical: critical conditions; issue that requires manager attention, possible problem.</td>
</tr>
<tr>
<td>err</td>
<td>3</td>
<td>Error: error conditions; issue that may require manager attention.</td>
</tr>
<tr>
<td>warning</td>
<td>4</td>
<td>Warning: warning conditions; normal notification of an event, not serious or particularly important.</td>
</tr>
<tr>
<td>notice</td>
<td>5</td>
<td>Notice: normal but significant condition; useful information, can be ignored during normal operation.</td>
</tr>
<tr>
<td>info</td>
<td>6</td>
<td>Informational: informational messages; generally unimportant everyday events.</td>
</tr>
<tr>
<td>debug</td>
<td>7</td>
<td>Debug: debug-level messages; extremely detailed (possibly high-volume) debugging information. Debug messages are only generated when debugging for a particular feature is enabled using the debug commands for that feature.</td>
</tr>
</tbody>
</table>

Log outputs

The following types of logging output are available:

- "Buffered log" on page 5
- "Permanent log" on page 6
- "Terminal log" on page 7
- "Console log" on page 8
- "Host log (for syslog)" on page 8
- "Email log" on page 9
- "External logging" on page 10

The buffered log is a file stored in RAM on the device. Because it is stored in RAM its content does not survive a reboot of the device. A device can only have one instance of the buffered log. The buffered log is enabled by default and has a filter to include messages with a severity level of 'notifications' and above.

The buffered log can be enabled or disabled using the commands:

```
awplus#configure terminal
awplus(config)#log buffered
awplus(config)#no log buffered
```
Additional filters can be added and removed using the commands:

```plaintext
awplus(config)#log buffered {exclude|facility|level|msgtext|program|size}
awplus(config)#no log buffered {exclude|facility|level|msgtext|program|size}
```

The following log buffered commands are available:

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>show log</td>
<td>Display the entire contents of the buffered log.</td>
</tr>
<tr>
<td>show log tail</td>
<td>Display the 10 most recent entries in the buffered log.</td>
</tr>
<tr>
<td>show log tail &lt;10-250&gt;</td>
<td>Display a specified number of the most recent entries in the buffered log.</td>
</tr>
<tr>
<td>show log config</td>
<td>Display the configuration of all log outputs.</td>
</tr>
<tr>
<td>log buffered size</td>
<td>Specify the amount of memory the buffered log may use.</td>
</tr>
<tr>
<td>clear log</td>
<td>Remove the contents of the buffered log and permanent log.</td>
</tr>
<tr>
<td>clear log buffered</td>
<td>Remove the contents of the buffered log only.</td>
</tr>
<tr>
<td>copy buffered-log</td>
<td>Copy the contents of the buffered log to a destination file in a different external or internal location. This command is available from 5.4.7-1.1 onwards.</td>
</tr>
<tr>
<td>default log buffered</td>
<td>Restore the buffered log to its default configuration.</td>
</tr>
</tbody>
</table>

**Permanent log**

The permanent log is a file stored in NVS on the device, unless the device has no NVS, in that case it is stored in Flash. The content of the permanent log is retained over a reboot.

On IE200-6 Series switches, files in NVS persist over a device restart but do not persist over a power cycle.

A device can only have one instance of the permanent log. The permanent log is enabled by default and has a filter to include messages with a severity level of “warning” and above.

The permanent log can be enabled or disabled using the commands:

```plaintext
awplus#configure terminal
awplus(config)#log permanent
awplus(config)#no log permanent
```

The following log permanent commands are available:

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>show log permanent</td>
<td>Display the entire contents of the permanent log.</td>
</tr>
<tr>
<td>show log permanent tail</td>
<td>Display the 10 most recent entries in the permanent log.</td>
</tr>
<tr>
<td>show log permanent tail &lt;10-250&gt;</td>
<td>Display a specified number of the most recent entries in the permanent log.</td>
</tr>
<tr>
<td>show log config</td>
<td>Display the configuration of all log outputs.</td>
</tr>
<tr>
<td>log permanent size</td>
<td>Specify the amount of memory the permanent log may use.</td>
</tr>
</tbody>
</table>
Terminal logging displays all log messages on the console as they occur. By default this includes messages at informational and debugging severity level. It can be useful for troubleshooting but can also result in large numbers of messages displaying on the console.

The terminal log can be enabled using the commands:

```
awplus#configure terminal
awplus(config)#debug <protocol> [<parameter>]
awplus(config)#exit
awplus#terminal monitor
```

To turn off terminal logging use the command:

```
awplus#terminal no monitor
```

From 5.4.8-0.2 onwards, you can also use the command:

```
awplus#no terminal monitor
```

To limit the terminal monitor output, use the following commands:

1. First remove the default filter:

   ```
   awplus(config)#no log monitor level debugging
   ```

2. Then add a filter that describes the messages you wish to see, for example OSPF:

   ```
   awplus(config)#log monitor program ospf
   ```

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>clear log</td>
<td>Remove the contents of the buffered log and permanent log.</td>
</tr>
<tr>
<td>clear log permanent</td>
<td>Remove the contents of the permanent log only.</td>
</tr>
<tr>
<td>copy permanent-log</td>
<td>Copy the contents of the permanent log to a destination file in a different external or internal location. This command is available from 5.4.7-1.1 onwards.</td>
</tr>
<tr>
<td>default log permanent</td>
<td>Restore the permanent log to its default configuration.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>default log terminal</td>
<td>Restores the default settings for log messages sent to the terminal when a log terminal command is issued. By default all messages are sent to the console when a log terminal command is issued.</td>
</tr>
<tr>
<td>terminal (filter)</td>
<td>Creates a filter to select messages to be sent to all consoles when the log terminal command is given. Selection can be based on the priority/severity of the message, the program that generated the message, the logging facility used, a sub-string within the message or a combination of some or all of these.</td>
</tr>
</tbody>
</table>
**Console log** This command configures the device to send log messages to consoles. The console log is configured by default to send messages to the device’s main console port. Terminal log and console log cannot be set at the same time. If console logging is enabled then the terminal logging is turned off.

The console log can be enabled or disabled using the commands:

```markdown
awplus# configure terminal
awplus(config)# log console
awplus(config)# no log console
```

The following log console commands are available:

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>default log console</td>
<td>Restores the default settings for log messages sent to the terminal when a log console command is issued. By default all messages are sent to the console when a log console command is issued.</td>
</tr>
<tr>
<td>console (filter)</td>
<td>Creates a filter to select messages to be sent to all consoles when the log console command is given. Selection can be based on the priority/severity of the message, the program that generated the message, the logging facility used, a sub-string within the message or a combination of some or all of these.</td>
</tr>
<tr>
<td>log console exclude</td>
<td>Removes the contents of the buffered log (and permanent log if it exists).</td>
</tr>
<tr>
<td>show log config</td>
<td>Displays information about the logging system. This includes the configuration of the various log destinations, buffered, permanent, syslog servers (hosts) and email addresses. This also displays the latest status information for each of these destinations.</td>
</tr>
</tbody>
</table>

**Host log** (for syslog) A host log sends log messages to a remote syslog server. A device may have many syslog hosts configured.

The host log can be enabled or disabled using the commands:

```markdown
awplus# configure terminal
awplus(config)# log host <ipv4-addr> [secure]
awplus(config)# log host <ipv6-addr>
awplus(config)# no log host <ipv4-addr>|<ipv6-addr>
```

where: `<ipv4-addr>` or `<ipv6-addr>` is the IP address of the remote syslog server.

The secure parameter is an optional setting used to create a secure log destination. It is only available for IPv4 hosts. For details of how to configure it, see “Syslog over TLS” in the PKI Feature Overview and Configuration Guide.

There are no default filters associated with host outputs when they are created. Filters can be added and removed using various parameters of the log host command.
It is not possible to view the log messages sent to this type of output as they are not retained on the device. They must be viewed on the remote device.

The other host log commands are:

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>show log config</td>
<td>Displays the configuration of all log outputs.</td>
</tr>
<tr>
<td>log host time</td>
<td>Adjust the time information in messages to a time zone other than the one configured on this device.</td>
</tr>
<tr>
<td>default log host &lt;ip-address&gt;</td>
<td>Restores the device default settings for log sent to a remote syslog server.</td>
</tr>
<tr>
<td>log facility</td>
<td>Specifies an outgoing syslog facility. This determines where the syslog server will store the log messages. (Available with 5.4.6-0.1 and later.)</td>
</tr>
<tr>
<td>log host source</td>
<td>Specifies a source interface or IP address for the device to send syslog messages from. This is useful if the device can reach the syslog server via multiple interfaces or addresses and you want to control which interface/address the device uses. (Available with 5.4.6-0.1 and later.)</td>
</tr>
<tr>
<td>log host startup-delay</td>
<td>Changes the delay between the device booting up and it attempting to connect to remote log hosts. This delay period allows time for network connectivity to the remote host to be established. During this period, the device buffers log messages and sends them once it has connected to the remote host. You can change the delay period and the number of messages buffered.</td>
</tr>
</tbody>
</table>

**Email log**

An email log sends log messages to an email address. A device may have many email logs configured.

The email log can be enabled or disabled using the commands:

```
awplus#configure terminal
awplus(config)#log email <email-address> {exclude|facility|level|msgtext|program|time}
awplus(config)#no log email <email-address> {exclude|facility|level|msgtext|program|time}
```

where `<email-address>` is the destination email address.

It is not possible to view the log messages sent to this type of output as they are not retained on the device. They must be viewed by the email recipient.
The other email log commands are:

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>show log config</td>
<td>Displays the configuration of all log outputs.</td>
</tr>
<tr>
<td>log email time</td>
<td>Adjusts the time information in messages to a time zone other than the one configured on this device.</td>
</tr>
<tr>
<td>default log email &lt;email-address&gt;</td>
<td>Restores the device's default settings for log messages sent to an email address.</td>
</tr>
</tbody>
</table>

Note: An email server and “from” address must be configured on the device in order for email logs to work:

- mail from <email-address>
- mail smtp <ip-address> | <domain-name>

Where the <email-address> is the “From” field on the sent email, and the <ip-address> or <domain-name> is the email’s destination SMTP server. Specifying the server by using its domain name is only available from software version 5.4.7-1.1 onwards.

Email logs are sent in batches of approximately 20 messages and have the subject line “Log messages”.

Note: For more information about email and mail, see the Mail (SMTP) Feature Overview and Configuration Guide.

External logging sends syslog messages to a file on a USB memory device or SD card. It is available from software version 5.4.7-1.1 onwards.

External logging can be enabled or disabled using the commands:

```
awplus#configure terminal
awplus(config)#log external <filename>
```

For example, to save messages to a file called “messages.log” in a directory called “log” on a USB stick, use the command:

```
awplus(config)#log external usb:/log/messages.log
```

If the file does not already exist on the memory device, it (and any specified subdirectory) will be automatically created. If the file already exists, messages are appended to it.

We strongly recommend using ext3 or ext4 as the file system on the external memory device. These file systems have a lower risk of file corruption occurring if the switch or firewall loses power.

You should also unmount the memory device before removing it from the switch or firewall, to avoid corrupting the log file. To unmount the device, use the unmount command.

If you are using this on a VCStack, each stack member needs to have its own external memory device. Enabling or disabling external logging enables or disables it on all stack members.
The other external log commands are:

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>show log config</td>
<td>Displays the configuration of all log outputs. This lets you see if the external logging is functioning by checking that the status is enabled.</td>
</tr>
<tr>
<td>show log external</td>
<td>Displays the contents of the external log.</td>
</tr>
<tr>
<td>log external size &lt;50-4194304&gt;</td>
<td>Configures the total amount of size that the external log is permitted to use, in kilobytes. The maximum possible depends on the memory device's file system. The default maximum size is 50 kBytes. Note that if you are rotating between multiple files, this is the maximum size of all files, not of each individual file. For example, if you are rotating between 2 files (log external rotate 1), each file will have a maximum size of 25 kBytes by default.</td>
</tr>
<tr>
<td>log external rotate &lt;0-255&gt;</td>
<td>Configures the number of files that the external log can rotate through. For example, the diagram below shows how setting rotate to 2 makes the device rotate through 3 files. The default is 1, which rotates between the initial file and 1 additional file (e.g. messages.log and messages.log.1). Note that if you set rotate to 0, and the external log file becomes full, then the device deletes the full log file and creates a new (empty) file of the same name to save messages into. For this reason, we recommend setting rotate to at least 1.</td>
</tr>
<tr>
<td>clear log external</td>
<td>Delete the external log file.</td>
</tr>
<tr>
<td>default log external</td>
<td>Restores the device’s default settings for external logging.</td>
</tr>
</tbody>
</table>
Filtering log message output

You can create a filter to select messages to be sent to each of the log output types. You can filter on the priority/severity level of the message, the program that generated the message, the logging facility used, a sub-string within the message, or a combination of some or all of these.

For example the command syntax for the permanent log is:

```
log permanent [level <level>] [program <program-name>] [facility <facility>] [msgtext <text-string>]
```

Configuring filters to include messages in different types of logging output

Here are two examples of configuring filters to include messages into different types of logging output:

1. To save all messages in the buffered log if they are generated by EPSR and have a severity of informational or higher, use the following commands:

   ```
   awplus#configure terminal
   awplus(config)#log buffered level informational program epsr
   ```

2. To save a specific part(s) of messages in the buffered log. For example:

   To filter only the OSPF message (SEND[Hello]: To 224.0.0.5 via vlan1:172.16.101.2, length 44) to appear on the console, or in "show log", use the following command:

   ```
   awplus(config)#log buffered program ospf msgtext To 224.0.0.5 via vlan1:172.16.101.2, length 44
   ```

   Figure 1 below shows some unfiltered OSPF debug output, but by using the filter to include only the “SEND[Hello]...” text, we get the filtered output shown in Figure 2.

   **Figure 1:** Unfiltered section of debug

   ```
   12:11:03 x930 OSPF[1893]: IFSM[vlan1:172.16.101.2]: Hello timer expire
   12:11:03 x930 OSPF[1893]: SEND[Hello]: To 224.0.0.5 via vlan1:172.16.101.2, length 44
   12:11:03 x930 OSPF[1893]: -----------------------------------------------------
   12:11:03 x930 OSPF[1893]: Header
   12:11:03 x930 OSPF[1893]:   Version 2
   12:11:03 x930 OSPF[1893]:   Type 1 (Hello)
   12:11:03 x930 OSPF[1893]:   Packet Len 44
   12:11:03 x930 OSPF[1893]:   Router ID 9.9.9.9
   12:11:03 x930 OSPF[1893]:   Area ID 0.0.0.0
   12:11:03 x930 OSPF[1893]:   Checksum 0xd87a
   12:11:03 x930 OSPF[1893]:   AuType 0
   12:11:03 x930 OSPF[1893]:   Hello
   12:11:03 x930 OSPF[1893]:   NetworkMask 255.255.0.0
   12:11:03 x930 OSPF[1893]:   HelloInterval 10
   12:11:03 x930 OSPF[1893]:   Options 0x2 (-|-|-|E-|-|-|-|-|-|)
   12:11:03 x930 OSPF[1893]:   RtrPriority 1
   12:11:03 x930 OSPF[1893]:   RtrDeadInterval 40
   12:11:03 x930 OSPF[1893]:   DRouter 172.16.101.2
   12:11:03 x930 OSPF[1893]:   BDRouter 0.0.0.0
   12:11:03 x930 OSPF[1893]:   # Neighbors 0
   12:11:03 x930 OSPF[1893]: -----------------------------------------------------
   ```
Configuring filters to exclude messages from different types of logging output

With version 5.4.4-4.13 and later you can drop unwanted log messages. For example, you can drop low priority log messages that are overfilling the log files. Use this with caution, to avoid dropping important messages.

To configure the device to drop logs, specify the level, program, facility, or message text you want to drop, and use the `exclude` parameter to specify to drop them. The `exclude` parameter option is available for all types of log output with version 5.4.4-4.13 and later. For example, the syntax for the buffered log is:

```
log buffered exclude [level <level>] [program <program-name>] [facility <facility>] [msgtext <text-string>]
```

**Example**  
Consider a situation where you often see log messages such as the following, which are harmless, but may be frustrating:

```
CoreSw EXFX[3101]: DBG:exfx_rxtx_rxInvalidPktPrint 358: Invalid packet RX'ed dmac 001e.67a7.0501 smac eccd.6d7b.3d92 ethertype 0800
CoreSw EXFX[3101]: DBG:exfx_rxtx_rxInvalidPktPrint 358: Invalid packet RX'ed dmac 001e.67a7.0501 smac eccd.6d7b.3d92 ethertype 0800
CoreSw EXFX[3101]: DBG:exfx_rxtx_rxPktSourceSet 339: Packet RX'ed with no ifindex srcDevNum 1 srcPortNum 61 vlan 1 cpuCode 473
```

You can filter these out by either filtering on the message text, or filtering on the program EXFX combined with level of "error":

- Filtering on the message text:

  You can filter out these messages by specifying the message text. The following examples use a `msgtext` filter to exclude logs from both the permanent log (`show log permanent`), and the buffered log (`show log`).

  ```
  awplus(config)#log permanent exclude msgtext exfx_rxtx_rxInvalidPktPrint
  awplus(config)#log permanent exclude msgtext exfx_rxtx_rxPktSourceSet
  ```

The filter has put a specific debug message into the log, without you having to turn on terminal monitor, and without flooding the console or log with unneeded debug messages.
Note that if you only base your filter on the message text, you will filter logs containing the specified text from all programs and severity levels.

- Filtering on program **EXFX** and level **error**, as well as specifying the message text:

Specifying the program and level as well as the message text ensures you only exclude the messages you actually intend to. We recommend using this instead of a filter that only specifies the message text, in case logs from other programs contain the same message text.

Restoring default settings

The following commands will restore logging configuration to defaults for each logging destination. This will undo and remove all filters for the specified destination. This is useful if there are a lot of log filters configured. Alternatively, you can remove specific configuration by using the standard `no` configuration command.

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>default log buffered</td>
<td>Restores the buffered log stored in RAM to its default configuration. By default the size of the buffered log is 50 kB and it accepts messages with a severity level of notifications and above.</td>
</tr>
<tr>
<td>default log permanent</td>
<td>Restores the default settings for the permanent log stored in NVS. By default, the size of the permanent log is 50 kB and it accepts messages with a severity level of warnings and above.</td>
</tr>
<tr>
<td>default log host &lt;ip-addr&gt;</td>
<td>Restores the device default settings for logs sent to a remote syslog server. By default no filters are defined for remote syslog servers. This command also restores the remote syslog server time offset value to local (no offset).</td>
</tr>
<tr>
<td>default log console</td>
<td>Restores the console log to its default configuration. By default all messages are sent to the console when a log console command is issued.</td>
</tr>
<tr>
<td>default log email</td>
<td>Restores the email log to its default configuration. By default no filters are defined for email addresses.</td>
</tr>
<tr>
<td>default log external</td>
<td>Restores the external log to its default configuration. By default, the size of the external log is 50 kB, it rotates through 1 additional file, and it accepts messages with a severity level of notices and above.</td>
</tr>
</tbody>
</table>
Activating Configuration Scripts with Log Message Triggers

This feature is supported from AlliedWare Plus version 5.4.7-2.1.

You can configure a trigger to activate a configuration script when a particular string is generated in log messages of severity level notice or higher. The log message string can be filtered by including regular expressions (PCRE). This section describes the general steps to configure a trigger to activate by a log message, and some simple syntax and examples for using regular expressions to match log messages for triggers.

Configuring a log message trigger

**Step 1: Create a command script**
Create a command script with the commands you would like executed when the trigger conditions are met. Either create a script on a PC then load it onto your device using the `copy (URL)` command, or create the command script directly on the device using the CLI, using the command:

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

Note that any command executed by the script will generate a log message with level notice, and will include [SCRIPT] before the command string. Therefore, if something in the script matches the configured log message trigger string, it will retrigger indefinitely.

**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 to log**
The trigger type determines how the trigger is activated. To set the trigger to activate if a particular string is generated in a log message, 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 16). Triggers are only activated by log messages of severity level notice (5) or higher. Log type triggers are limited to activating at most once per second.

**Step 4: Add the scripts to the trigger**
You can add up to five 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>
```
Step 5: 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 6: Verify the trigger’s configuration

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

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

Triggers can also be configured to activate only on particular days, particular times during the day, for a limited number of repetitions, and for other types of events. For more information about configuring triggers, see the Triggers Feature Overview and Configuration Guide and the ‘Trigger Commands’ chapter in the Command Reference for your device.

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.

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>Use for any character.</td>
</tr>
<tr>
<td>\d</td>
<td>Use for any number.</td>
</tr>
<tr>
<td>*</td>
<td>Use for 0 or more occurrences of the preceding element.</td>
</tr>
<tr>
<td>+</td>
<td>Use for 1 or more occurrences of the preceding element.</td>
</tr>
<tr>
<td>()</td>
<td>Use parentheses for grouping.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>[]</td>
<td>Use brackets to enclose a set of characters.</td>
</tr>
</tbody>
</table>

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}
  ```