Collect and Manage Trace Logs with Unified Logging Enhancement

Contents

Introduction

This document describes the new unified logging enhancement for a seamless experience to collect and management Tracelogs of the sytem.

Overview

Practical Purposes:

- Troubleshooting. When a chassis encounters a problem, the data within trace files can be invaluable for identifying and resolve the issue.
- Debugging. The outputs from trace files can offer users a more granular perspective on the actions and operations of the system.

How they Work

- The tracing feature records the details of internal events occurring within the chassis. Periodically, trace files containing the complete trace output for a module are generated and refreshed, and these files are kept in the **tracelog** directory.
- You can free up space on the file system. Trace files can be deleted from this directory without affecting the performance of the device.

- You can transfer the trace logs to alternative locations. You can use FTP, TFTP, and so on, to copy the files to analyze them or to upload them to a case opened with Technical Assistance Center (TAC).
- You cannot disable trace logs but you can change the trace level to determine how much information you want to collect for each module.

Trace Levels

Trace levels dictate the volume of information retained in the trace buffer or file. All the available tracing levels are next and explain the kinds of messages that are logged at each level.

Emergency--> System is unstable/unusable.

Error--> An event that results in a minor loss of functionality without automatic resolution, representing an unanticipated issue that can not affect operations immediately but could have future consequences.

Warning--> A problem that can potentially be resolved automatically, or a condition that can lead to a loss of functionality if not promptly investigated and addressed.

Notice--> The standard log level set for modules. This level captures important events that occur within the system.

Info--> Only informational messages. Provides additional information about significant events that are relevant to the system or its features.

Debug--> Provides debugging logs.
Verbose--> Provides second level of debugging logs.
Noise--> Maximum possible messages are logged.

View Current Trace Levels

You can see and change the trace level of any module with the command **show platform software trace level.**

This command shows the Tracing Level of forwarding manager processes on the active RP.

Router#**show platform software trace level forwarding-manager rp active**

This is the output:

```
Module Name                      Trace Level
----------------------------------------------
acl                              Notice
active-identity                  Notice
alg                              Notice
appnav-controller                Notice
aps                              Notice
bcrpgc                           Informational
bfd                              Notice
bier                             Notice
<SNIP>
```

Modify Trace Level

You can modify a trace level for a specific module or all modules in a process. To do that you can use the command **set platform software trace**.

This command **set platform software trace chassis-manager f0 cman_fp warning** changes the trace level for **cman_fp** in the chassis manager of the ESP in the slot 0 to **warning level**.

You can validate the change with this command **show platform software trace level chassis-manager f0**

This is the output:

```
Module Name                      Trace Level
-----------------------------------------------
bcrpgc                           Informational
bipc                             Notice
bsignal                          Notice
btrace                           Notice
btrace_ra                        Notice
cdllib                           Notice
chasfs                           Notice
cman_fp                          Warning
```

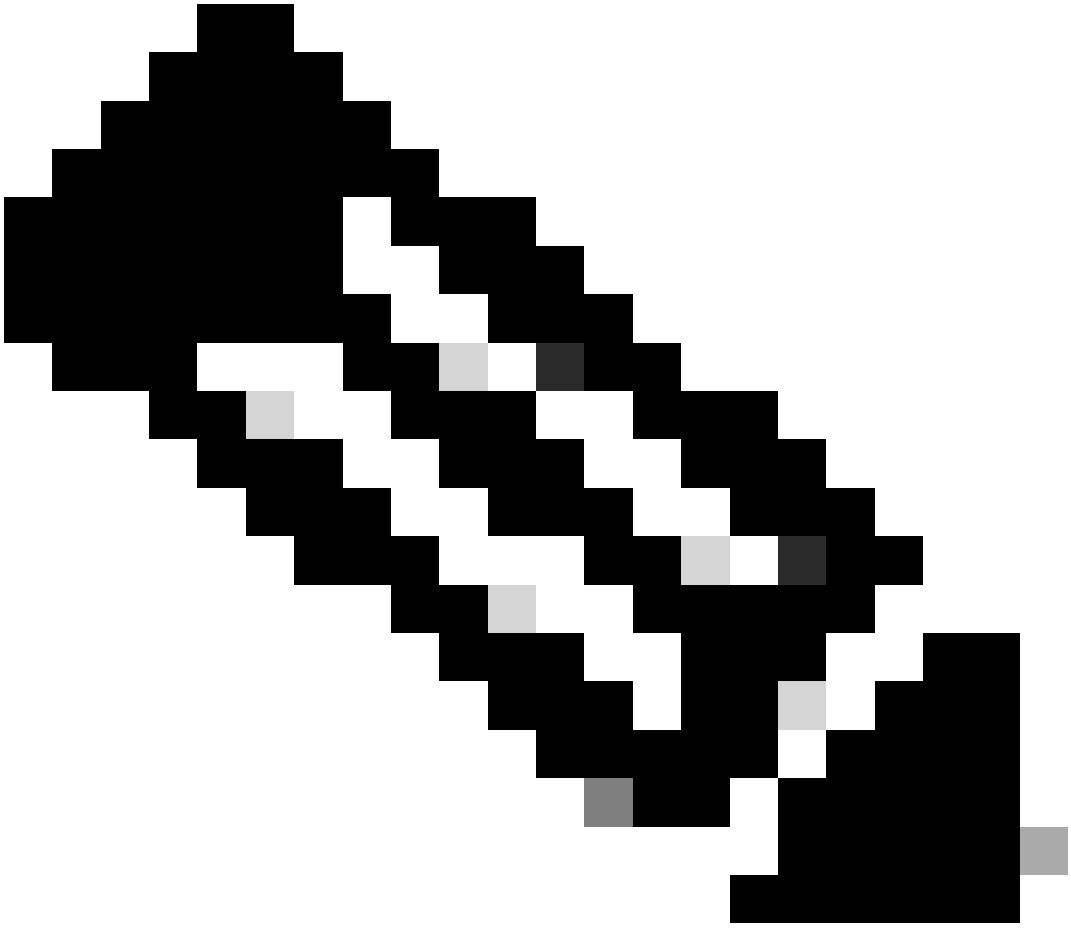
New Trace Options

Starting in 16,8 , Cisco introduces the Unified Logging enhancement. The goal is to create a seamless logging experience for the user between the IOS logging and other process logging systems. Logs from both systems can be merged and displayed in time order, This makes easer for you to troubleshoot issues in the system.

Display the Tracelogs for different process

The command **show logging process** can be used to display the content of the tracelogs generated by the specified processes. Logs from the buffer and tracelogs directory can be included in the output.

There is also support for partial process name the process name is accepted as a **word**by the parser.



Note: Process name must match (partially or completely) the name of the tracelog, otherwise, a mismatch can occur resulting in no traces being displayed.

The command show logging process fman can combine the **fman-rp** and **fman-fp** logs.

```
Router#show logging process fman
Displaying logs from the last 0 days, 0 hours, 10 minutes, 0 seconds
executing cmd on chassis local ...
Unified Decoder Library Init .. DONE
Found 1 UTF Streams
<snip>
2024/05/22 19:01:01.347466887 {fman_rp_R0-0}{255}: [source] [11941]: (ERR): ipc(mqipc/iosd/iosd-fman):U
2024/05/22 19:00:50.246774567 {fman_fp_image_F0-0}{255}: [btrace] [13616]: (note): Btrace started for p
2024/05/22 19:00:50.246777079 {fman_fp_image_F0-0}{255}: [btrace] [13616]: (note): File size max used f
```

Perform Multiple Process Support

The "show logging process" CLI command now allows you to specifying multiple process names separated by commas using the 'process' keyword. Merged tracelog displaying logs only from the specified processes, such as **sman** and **ios** processes**.**

```
Router#show logging process sman,ios
executing cmd on chassis 1 ...
Collecting files on current[1] chassis.
```

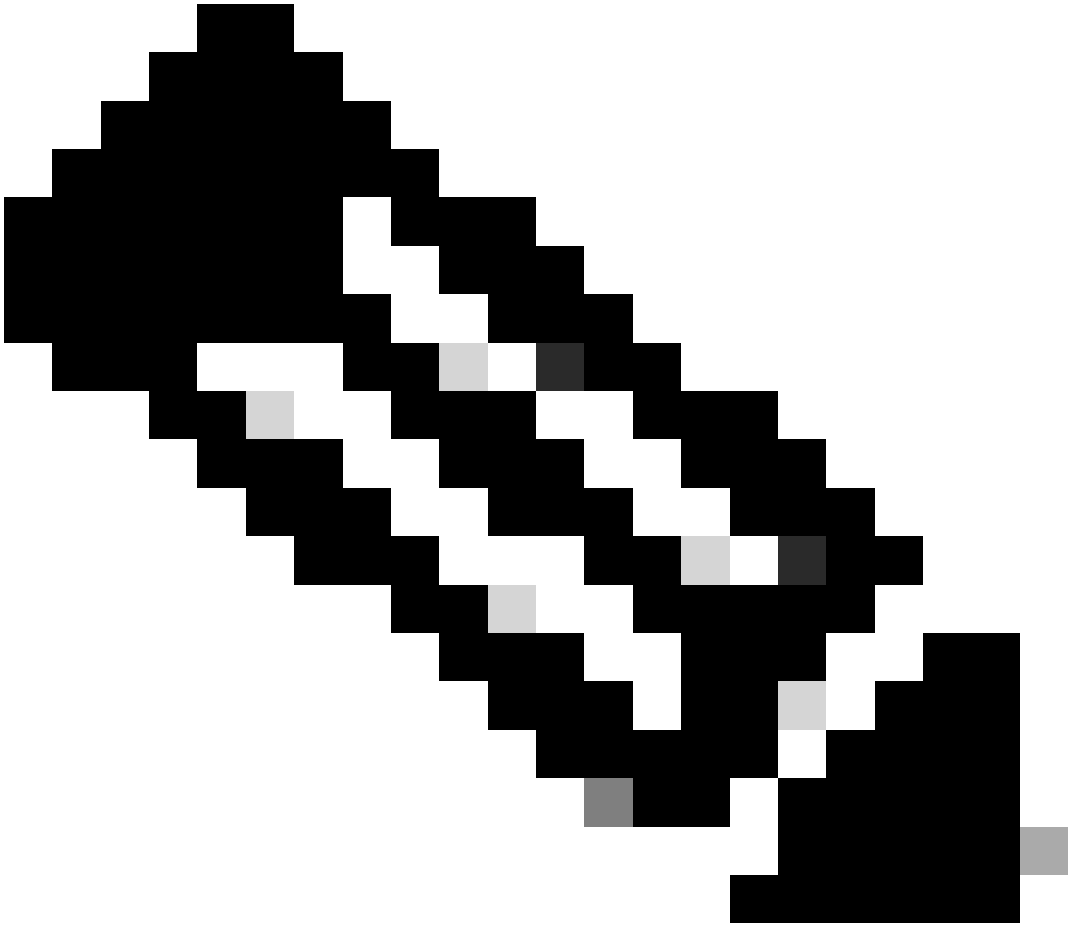
Time Window Options

All traces can have **timestamps** in a local timezone if any are configured. If not, **timestamps** is used in "Coordinated Universal Time" (UTC) but you can switch the time zone for the logs between **Local** and **UTC** with the command **set logging timezone <local | UTC>.**

The command **show logging** CLI only displays the last **10 minutes** of logs from the current time by default.

The **start last** keyword can be used to expand the time window as per individual needs.

```
Router#show logging process btman start last ?
  <0-4294967295>  interval (default seconds)
  boot            system boot time
  clear           display all logs since last "clear logging"
  marker          selects latest matching marker from list to start displaying
                  logs from
```



Note: If you choose a numeric value from previous command, then you can specify days, hours, minutes or seconds as next option.

The **end last** keyword options have been added to be used in conjunction with **start last** to specify the end of the time window. When both **start last** and **end last** options are used, only logs within the window are collected. Without **end last** option, log collection defaults to the current time as the end time.

This is an example setting a window between the last two hours and last one hour:

```
Router#show logging process btman start last 2 hours end last 1 hours
Displaying logs from the last 0 days, 2 hours, 0 minutes, 0 seconds
End time set to show logs before last 0 days, 1 hours, 0 minutes, 0 seconds
executing cmd on chassis 1 ...
Collecting files on current[1] chassis.
```

Show Logs Under Specific Log Level

You can show logs only for specific level:

```
Router#show logging process wncd level ?
  debug    Debug messages
  error    Error messages
  info     Informational messages
  notice   Notice messages
  verbose  Verbose debug messages
  warning  Warning messages
```

This is an example of the logs btman under error level **notice**:

```
Router#show logging process btman level notice
Logging display requested on 2024/07/24 06:20:23 (UTC) for Hostname: [Router], Model: [ASR1002-HX

Displaying logs from the last 0 days, 0 hours, 10 minutes, 0 seconds
executing cmd on chassis local ...
Unified Decoder Library Init .. DONE
Found 1 UTF Streams

2024/07/24 06:10:59.533374335 {btman_RO-0}{255}: [utm_main] [5809]: (note): Inserted UTF(2) HT(old):dro
2024/07/24 06:10:59.695395289 {btman_RO-0}{255}: [utm_wq] [5809:15578]: (note): Inline sync, enqueue BT
```

Show Logs From Time-Stamp

You can show logs from specific time-stamp in UTC like this "2017/02/10 14:41:50.849425" This is an example:

```
Router#show logging process wncd start timestamp "2024/07/24 05:36:45.849425"
Logging display requested on 2024/07/24 06:39:15 (UTC) for Hostname: [Router], Model: [ASR1002-HX

executing cmd on chassis local ...
Unified Decoder Library Init .. DONE
Found 1 UTF Streams

Filter policy: Done with UTM processing
```

Show Logs Between Two Time-Stamp

You can display tracelogs between a time window by adding **start timestamp** and **end timestamp.** This is an example with 1 hour of maintenance window:

```
Router#show logging process wncd start timestamp "2024/07/24 05:36:45.849425" end timestamp "2024/07/24
Logging display requested on 2024/07/24 06:39:15 (UTC) for Hostname: [Router], Model: [ASR1002-HX

executing cmd on chassis local ...
```

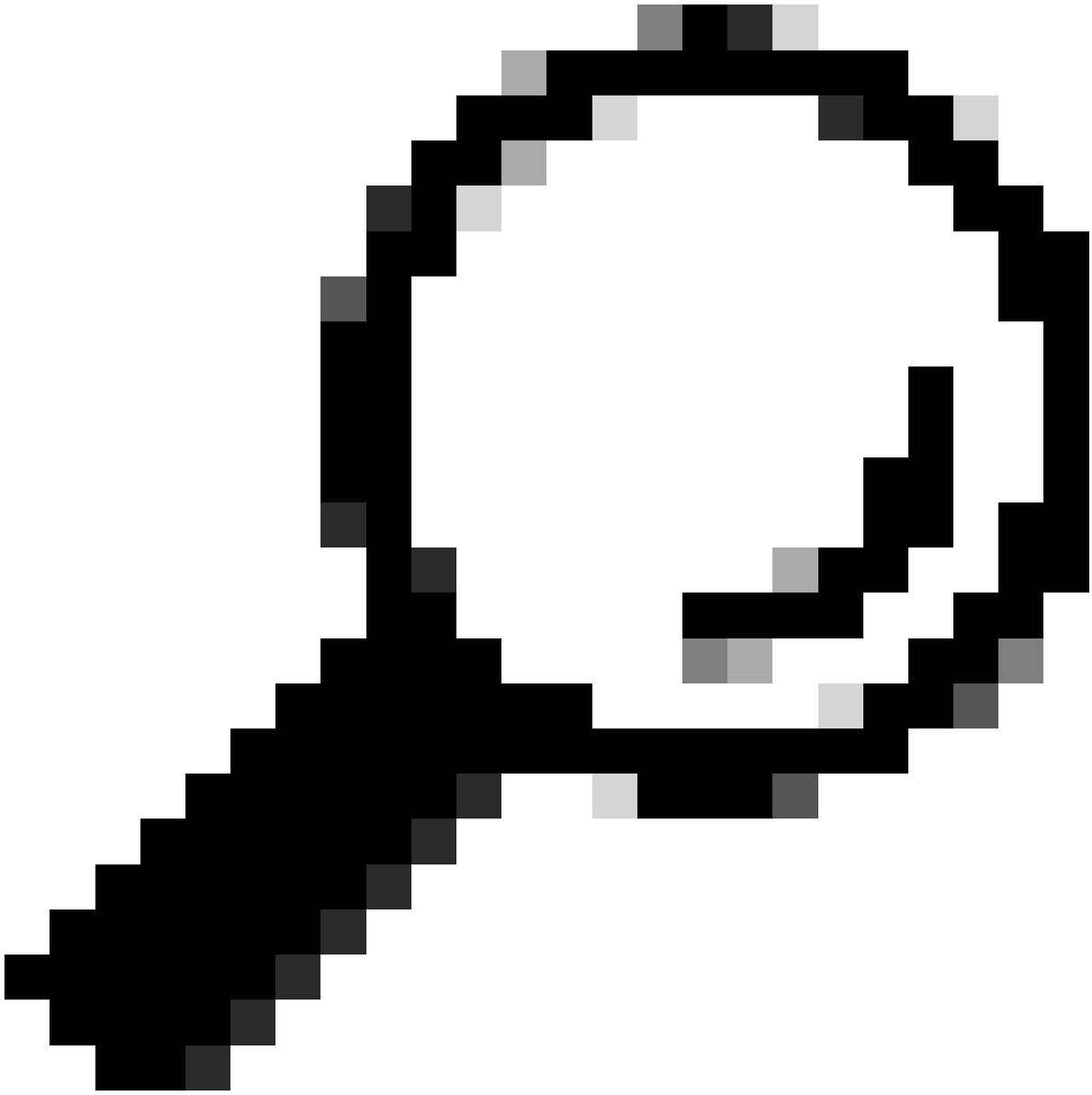
```
Unified Decoder Library Init .. DONE
Found 1 UTF Streams

Filter policy: Done with UTM processing
```

Perform Live Logging

You can monitor the logs generated in real-time for a process or profile. Logs are showing as they are generated.

```
Router#monitor logging process cman ?
  <0-25>    instance number
  filter    specify filter for logs
  internal  select all logs. (Without the internal keyword only customer
            curated logs are displayed)
  level     select logs above specific level
  metadata  CLI to display metadata for every log message
  module    select logs for specific modules
  <cr>      <cr>
```

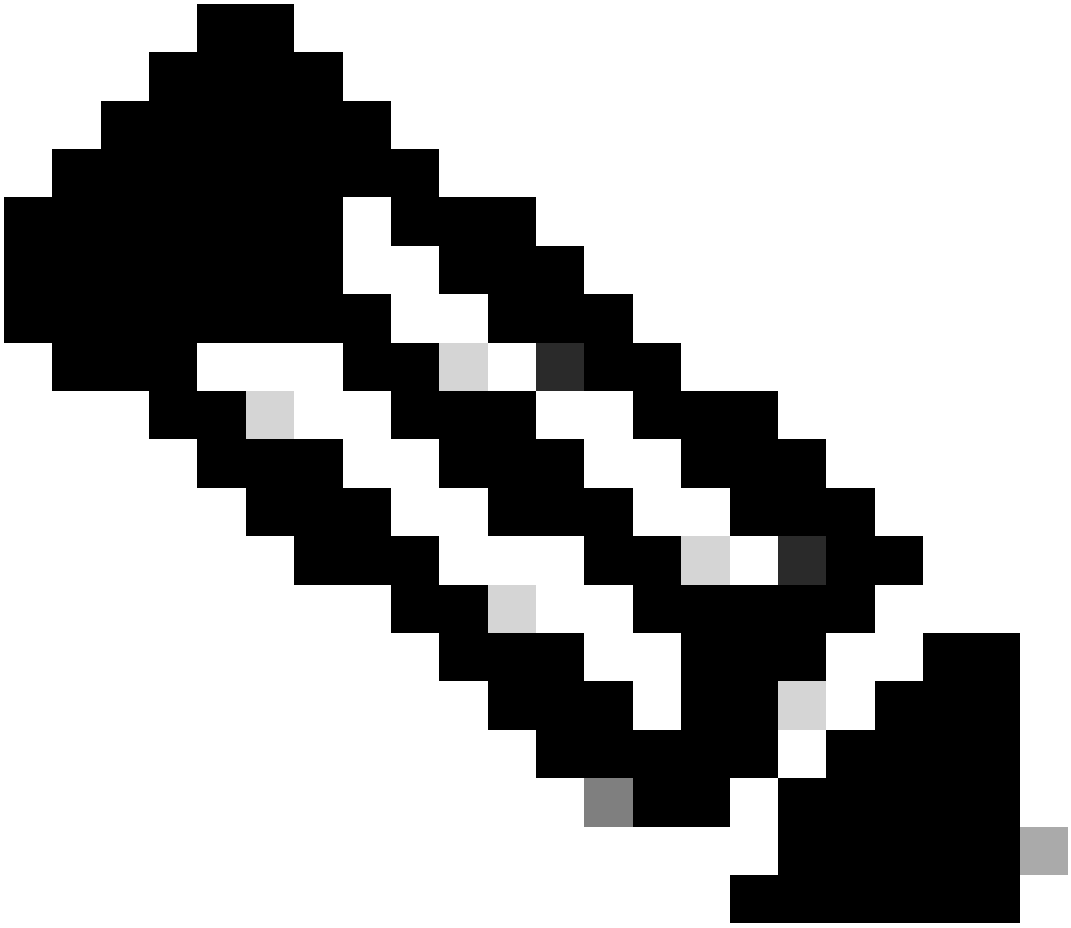


Tip: To escape, or exit, live logging mode issue a CTRL-C.

Use Predefined Log Profiles

Live logging offers built-in profiles that can be easily applied. This eliminates the need for the user to be familiar with the underlying process log files that constitute the feature. The supported profiles include all, file, wireless, sdwan, netconf-yang, restconf, install, hardware-diagnostics.

The profile can be used to with **show logging** or the **monitor logging** command.



Note: The show logging profile option only displays logs from the buffer and does not include logs from the tracelogs directory.

```
SCP_Test#show logging profile ?
  all                   all processes
  file                  show logs for specific profile file
  hardware-diagnostics  hardware diagnostics specific processes
  install               Install specific processes
  netconf-yang          netconf-yang specific processes
  restconf              restconf specific processes
  sdwan                 SDWAN specific processes
  wireless              Wireless specific processes



Router#monitor logging profile ?
  all                   all processes
  file                  show logs for specific profile file
  hardware-diagnostics  hardware diagnostics specific processes
  install               Install specific processes
  netconf-yang          netconf-yang specific processes
```

```
  restconf              restconf specific processes
  sdwan                 SDWAN specific processes
  wireless              Wireless specific processes
```

In 17.12+ the stats are included by default at the end of "show logging <process/profile/file> ..". Stats shows the number of trace messages decoded at each severity level were added to the existing decoder stats. The level counts are for rendered traces only.

```
2024/07/24 04:26:41.710239127 {btman_RO-0}{255}: [utm_wq] [5806:15568]: (note): Inline sync, enqueue BT
2024/07/24 04:26:41.759114843 {btman_RO-0}{255}: [utm_wq] [5806]: (note): utm delete /tmp/rp/trace/IOSR
============================================================
======= Unified Trace Decoder Information/Statistics ======
============================================================
----------------- Decoder Input Information --------------
============================================================
 Num of Unique Streams .. 1
 Total UTF To Process ... 1
 Total UTM To Process ... 89177
 UTM Process Filter ..... btman
 MRST Filter Rules ...... 1
============================================================
----------------- Decoder Output Information --------------
============================================================
 First UTM TimeStamp ............... 2024/07/24 02:51:45.623542304
 Last UTM TimeStamp ................ 2024/07/24 04:26:48.710794233
 UTM [Skipped / Rendered / Total] .. 89047 / 130 / 89177
 UTM [ENCODED] ..................... 130
 UTM [PLAIN TEXT] .................. 0
 UTM [DYN LIB] ..................... 0
 UTM [MODULE ID] ................... 0
 UTM [TDL TAN] ..................... 0
 UTM [APP CONTEXT] ................. 0
 UTM [MARKER] ...................... 0
 UTM [PCAP] ........................ 0
 UTM [LUID NOT FOUND] .............. 0
 UTM Level [EMERGENCY / ALERT / CRITICAL / ERROR] .. 0 / 0 / 0 / 0
 UTM Level [WARNING / NOTICE / INFO / DEBUG] ....... 0 / 130 / 0 / 0
 UTM Level [VERBOSE / NOISE / INVALID] ............. 0 / 0 / 0
============================================================
```

Send Log Outputs To a File

You can use to-file keyword to create a file with the outputs of the command **show logging**. This example shows you how to send the trace logs of the process **btman** to a file named **btman_log.txt** in the **bootflash** file system:

```
Router#show logging process btman to-file bootflash:btman_log.txt
Logging display requested on 2024/07/25 03:49:41 (UTC) for Hostname: [Router], Model: [ASR1006-X

Displaying logs from the last 0 days, 0 hours, 10 minutes, 0 seconds
executing cmd on chassis local ...
Files being merged in the background, please check [/bootflash/btman_log.txt] output file
Unified Decoder Library Init .. DONE
```

```
unified trace decoder estimates: [1] number of files, [139913] number of messages
         that may be processed. Use CTRL+SHIFT+6 to break.
Found 1 UTF Streams
         2024-07-25 03:49:41.694987 - unified trace decoder estimate: processed 5%
         2024-07-25 03:49:41.701433 - unified trace decoder estimate: processed 10%
         2024-07-25 03:49:41.707803 - unified trace decoder estimate: processed 15%
         2024-07-25 03:49:41.714185 - unified trace decoder estimate: processed 20%
         2024-07-25 03:49:41.720592 - unified trace decoder estimate: processed 25%
         2024-07-25 03:49:41.726951 - unified trace decoder estimate: processed 30%
         2024-07-25 03:49:41.733306 - unified trace decoder estimate: processed 35%
         2024-07-25 03:49:41.739734 - unified trace decoder estimate: processed 40%
         2024-07-25 03:49:41.746114 - unified trace decoder estimate: processed 45%
         2024-07-25 03:49:41.752462 - unified trace decoder estimate: processed 50%
         2024-07-25 03:49:41.758864 - unified trace decoder estimate: processed 55%
         2024-07-25 03:49:41.765225 - unified trace decoder estimate: processed 60%
         2024-07-25 03:49:41.771582 - unified trace decoder estimate: processed 65%
         2024-07-25 03:49:41.777968 - unified trace decoder estimate: processed 70%
         2024-07-25 03:49:41.784330 - unified trace decoder estimate: processed 75%
         2024-07-25 03:49:41.790693 - unified trace decoder estimate: processed 80%
         2024-07-25 03:49:41.797099 - unified trace decoder estimate: processed 85%
         2024-07-25 03:49:41.803462 - unified trace decoder estimate: processed 90%
         2024-07-25 03:49:41.811411 - unified trace decoder estimate: processed 95%
         2024-07-25 03:49:41.822322 - unified trace decoder estimate: processed 100%
         2024-07-25 03:49:41.822335 - unified trace decoder : processing complete Result:[Success]
```

You can validate that file was created with the command **dir bootflash** and filtering the name of the file like this:

```
Router#dir bootflash: | include btman_log.txt
17      -rw-            26939  Jul 25 2024 03:49:41 +00:00  btman_log.txt
```