VSI OpenVMS
LAN Driver Tracing Guide

Operating Systems: VSI OpenVMS Alpha Version 8.4-2L1 or higher
VSI OpenVMS IA-64 Version 8.4-1H1 or higher
VSI OpenVMS x86-64 Version 9.2-3 or higher

VMS Software, Inc. (VSI)
Boston, Massachusetts, USA

VSI OpenVMS LAN Driver Tracing Guide

Legal Notice

Table of Contents

1. LAN Driver Tracing

The LAN drivers record trace data of various events during LAN driver operation, including user startup and shutdown, link changes, and transmit and receive errors. Transmit and receive packet data can also be recorded.

The packet data and other notable trace events can be written to a PCAP (Packet Capture) file for analysis using Wireshark.

Tracing is initiated at device configuration using default settings. These settings can be modified by setting or clearing certain flags in the LAN_FLAGS system parameter, or via LAN Control Program (LANCP) commands or LANACP device settings. The LAN_FLAGS bits that indicate these flags, and their respective functions, are as follows:

- Bit24 – Stop tracing when the trace buffer is full.
- Bit27 – Disable tracing, until overridden by a LANCP command or LANACP device settings. Overrides Bit29 and Bit30.
- Bit29 – Enable debug mode, which enables tracing of most events and uses a larger trace buffer.
- Bit30 – Enable all tracing, which enables tracing of most events.

The trace data can be displayed using the LAN extension for the System Dump Analyzer (SDA) and by using LANCP commands. For more information on using SDA, refer to the relevant section in the *VSI OpenVMS System Analysis Tools Manual* [https://docs.vmssoftware.com/vsi-openvms-system-analysis-tools-manual/#PART1].

2. LANCP Trace Commands

This section describes some LANCP commands that are useful for driver tracing. For further information on the LANCP utility and LANCP commands, refer to the relevant section in the *VSI OpenVMS System Management Utilities Reference Manual, Volume I: A–L* [https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-l/#d0e28651] or the LANCP online Help.

2.1. Setting Trace Parameters

The following command can be used to set trace parameters:

```
$ MCR LANCP SET DEVICE devname /[NO]TRACE=([NO]MASK=valuelist, [NO]SIZE=n,
[NO]STOP=valuelist,keywords) [/ALL]
```

The *valuelist* variable is a pair of values (high 32 bits, low 32 bits) and combine to a 64-bit mask. For example, (33,2) is a 64-bit mask 0x0000002100000002.

- The trace mask selects events to be traced.
- The stop mask selects events to stop tracing. Tracing stops 64 trace entries later. For example, if you select a transmit error and want to see what happens after the error. Tracing does not stop on the matched event, but continues for a few milliseconds so that you can see the aftermath.

The following keywords cover most events:

Keyword	Function
PAUSE	Stops tracing, but does not deallocate the trace buffer, allowing the user to look at the trace data without it being disturbed.
DEFAULT	Selects the default trace mask.
MALL/MNONE	MALL selects all trace mask bits. MNONE selects no trace mask bits.
SALL/SNONE	SALL selects all stop mask bits. SNONE selects no stop mask bits.
RTINT	Selects a common set of trace events; receive done, transmit issued, and interrupt.
RTFINT	Selects a common set of trace events; the same as RTINT, plus fork start and fork done.
PK	Selects full packet data, transmit and receive. Variants include SMALLPK, MEDIUMPK, and LARGEPK.

If no trace entries are specified by mask or keyword, transmit and receive full packet data is traced.

If keywords are supplied, only those bits are part of the trace mask. For example, **/TRACE=(NOMASK,PK)** is equivalent to **/TRACE=PK**.

There are keywords for all the other trace bits. In addition, you can specify a bit number, such as M00 for the one-second timer trace mask bit or S00 for the equivalent stop mask bit. The SMALL, MEDIUM, LARGE, and FULL packet keywords affect how much of each packet is saved in the trace data.

The keywords are listed in the LANCP trace output header (with the bit number and uppercase keyword). The active mask and stop bits are listed as M and S respectively in the columns next to the bit number. See the following example:

```
SystemName Device Trace Data DEVNAME (Device Description) (22-NOV-2025 19:46:20.26):
  Trace mask 00404000,00000000, stop mask 00000000,00000000, trace buffer size 2048 entries
    00000001,0    00 TIMER-One second timer       01000000,0    24 XLARGE-Transmit pk (lg)
    00000002,0    01 INTR-Interrupt               02000000,0    25 XFULL-Transmit pk (all)
    00000004,0    02 FSTART-Fork started          04000000,0    26 VCIACTION-VCI action
    00000008,0    03 FDONE-Fork done              08000000,0    27 LANACTION-LAN action
    00000010,0    04 FERROR-Fork error            10000000,0    28 LANOTHER-LAN other
    00000020,0    05 FSOFT-Fork soft error        20000000,0    29 RDL error
    00000040,0    06 STATE-State change           40000000,0    30 Segment
    00000080,0    07 UINIT-First user start       80000000,0    31 Bit31
    00000100,0    08 UCHANGE-User change          0,00000001    32 Link state
    00000200,0    09 USTOP-Last user stop         0,00000002    33 Transmit timeout
    00000400,0    10 SHUTDOWN-Device shutdown     0,00000004    34 Reset device
    00000800,0    11 RISSUE-Receive issued        0,00000008    35 Setup link
    00001000,0    12 RDONE-Receive done           0,00000010    36 Check link
    00002000,0    13 RERROR-Receive error         0,00000020    37 SET_MAC function
    00004000,0 M  14 RSMALL-Receive pk (sm)       0,00000040    38 Message
    00008000,0    15 RMEDIUM-Receive pk (md)      0,00000080    39 Long fork time
    00010000,0    16 RLARGE-Receive pk (lg)       0,00000100    40 Long transmit
    00020000,0    17 RFULL-Receive pk (all)       0,00000200    41 Long receive
    00040000,0    18 XISSUE-Transmit issued       0,00000400    42 PHY rw error
    00080000,0    19 XQUEUE-Transmit queued       0,00000800    43 Auto-negotiation
    00100000,0    20 XDONE-Transmit done          0,00001000    44 Receive error pkt
    00200000,0    21 XERROR-Transmit error        0,00002000    45 Long second
    00400000,0 M  22 XSMALL-Transmit pk (sm)      0,00004000    46 Other
    00800000,0    23 XMEDIUM-Transmit pk (md)
  MALL,MNONE,SALL,SNONE - M00..M63,S00..S63
  RTINT - INTR, RDONE, XISSUE, XDONE
  RTFINT - RTINT, FSTART, FDONE
  PK - RFULL, XFULL (also SMALLPK, MEDIUMPK, LARGEPK)
```

The size is the number of 32-byte trace entries to allocate for the trace buffer: the minimum is 512, and the maximum is 1 million. If this value is not specified, the size is not changed. The initial (default) size is 512. The size if trace entries are selected but no size has been specified is also 512.

The **/ALL** qualifier applies to the *devname*. The **EI/ALL** qualifier selects all EI devices, the **E/ALL** qualifier selects all Ethernet devices, and the **/ALL** qualifier selects all devices. The **EIA/ALL** qualifier selects all EIA devices; since there is only one EIA device, **EIA/ALL** is equivalent to **EIA**.

The **SET DEVICE/TRACE** command restarts tracing with a reinitialized trace buffer: tracing does not resume, it is restarted from scratch.

2.2. Extracting Trace Data for Wireshark

The following command can be used to extract trace data (for later analysis using Wireshark):

```
$ MCR LANCP SHOW DEVICE [devname] /TRACE [/SELECT=(valuelist,keywords)] [/REVERSE]
[/OUTPUTFILE=textfilename] [/PCAPFILE=pcapfilename] [/[NO]HEADER]
```

The PCAP file is in binary Wireshark libpcap format.

The **/ALL** qualifier is implied.

Trace output includes the trace header above, followed by a list of trace entries, like so:

```
1  27474.8703476        0   17:10:21.15  Transmit issued   xmt 0 rindex 96 plen 42 vtag 0000 vcrp 81929B00
    DA FF-FF-FF-FF-FF-FF  SA 08-00-27-73-13-92  PTY 08-06
    00000000 00000000 00000000 00000000 00000608 92137327 0008FFFF FFFFFFFF  ........'s...................  :0000
                                                0000 00000000 00000000  ..........                     :0020
4  27474.8704314      838   17:10:21.15  Interrupt         p1 C800C0 p2 0 p3 30 p4 0
5  27474.8704471      156   17:10:21.15  Fork started      p1 0 p2 0 p3 0 p4 0
6  27474.8704478        6   17:10:21.15  Transmit done     xmt 0 rindex 95 plen 42 vtag 0000 vcrp 81929B00
7  27474.8704618      140   17:10:21.15  Fork done         p1 0 p2 0 p3 0 p4 0
8  27474.8705240      622   17:10:21.15  Interrupt         p1 C800C0 p2 0 p3 31 p4 0
9  27474.8705361      121   17:10:21.15  Fork started      p1 0 p2 0 p3 0 p4 0
10  27474.8705532      170   17:10:21.15  Receive done      rcv 199 rindex 0 plen 64 vtag 0000 vcrp 81916000
    DA 08-00-27-73-13-92 SA 52-54-00-12-35-02 PTY 08-06
    0202000A 02351200 54520200 04060008 01000608 02351200 54529213 73270008  ..'s..RT..5.....RT..5.  :0000
    07F0B1FD 00000000 00000000 00000000 00000000 00000F02 000A9213 73270008  ..'s........................  :0020
13  27474.8705590       58   17:10:21.15  Fork done         p1 0 p2 0 p3 0 p4 0
```

The first column is the sequence number, which is the number of 32-byte chunks used. In this example, the first trace entry is a transmit packet that spans three 32-byte chunks. The next trace entry is sequence number 4.

The second and fourth columns are the current time in seconds since boot and system time, respectively. The third column is the number of 100 *n*-second time units since the last trace entry.

The fifth column is the name of the trace entry.

The remaining output is trace-entry-specific. There are four longwords listed as `p1`, `p2`, `p3`, and `p4`.

The **/SELECT** qualifier allows display of only the selected trace entries. These are selected using the same keywords as **SET DEVICE/TRACE**.

The **/HEADER** qualifier indicates that only the header information should be displayed, allowing users to see the trace mask and names of the trace entry types. The default is **/NOHEADER**.

2.3. Saving Trace Data

The following command can be used to save trace data:

```
$ MCR LANCP SHOW DEVICE devname /TRACE /DUMPFILE=filename [/PCAPFILE=pcapfilename]
```

The **/DUMPFILE** qualifier causes the trace data to be written to a dump file. The LAN SDA extension includes the **/DUMPFILE=*filename*** qualifier as well, so that trace data can be extracted from a dump (or the running system) using SDA.

```
$ MCR LANCP SHOW DEVICE devname /INPUTFILE=filename [/PCAPFILE=pcapfilename]
```

The **/INPUTFILE** qualifier redirects LANCP to read the trace data from the specified dump file rather than the device specified.

This command is used to display PCAP output for trace data extracted from a system dump file or trace data saved to a dump file for later perusal. The LAN SDA extension does not include PCAP output, so you would save the trace data to a dump file and then use LANCP to read the dump file and write the PCAP output.

2.4. Examples

1. The following commands start tracing on the device `EIA` and select full packet data. The trace buffer size is not changed. All four commands are synonymous:

   ```
   $ MCR LANCP SET DEVICE/TRACE EIA
   $ MCR LANCP SET DEVICE/TRACE=PK EIA
   $ MCR LANCP SET DEVICE/TRACE=(RFULL,XFULL) EIA
   $ MCR LANCP SET DEVICE/TRACE=MASK=(%x02020000,0) EIA
   ```

2. The following command starts tracing with a common trace request to look at interrupts and timing of transmit issue, completion, and receive completion. The trace buffer size is not changed:

   ```
   $ MCR LANCP SET DEVICE/TRACE=RTINT EIA
   ```

   The following command is equivalent to the previous one, but additionally selects fork processing:

   ```
   $ MCR LANCP SET DEVICE/TRACE=RTFINT EIA
   ```

3. The following commands start tracing as in the previous example, but also include "Receive issued" (receive buffer given to the device). The two commands are synonymous:

   ```
   $ MCR LANCP SET DEVICE/TRACE=(RTINT,RISSUE) EIA
   $ MCR LANCP SET DEVICE/TRACE=(RTINT,M11) EIA
   ```

4. The following command pauses tracing:

   ```
   $ MCR LANCP SET DEVICE/TRACE=PAUSE EIA
   ```

5. The following commands stop tracing and deallocate the trace buffer. The two commands are synonymous:

   ```
   $ MCR LANCP SET DEVICE/NOTRACE EIA
   $ MCR LANCP SET DEVICE/TRACE=SIZE=0 EIA
   ```

6. The following examples show various commands that are useful when displaying trace results.

   - Specify forward or reverse direction, respectively:

     ```
     $ MCR LANCP SHOW DEVICE/TRACE EIA
     $ MCR LANCP SHOW DEVICE/TRACE EIA/REVERSE
     ```

Note

In the LAN SDA extension, the equivalent commands are as follows:

```
SDA> LAN TRACE/DEVICE=EIA
SDA> LAN TRACE/DEVICE=EIA/REVERSE
```

- Specify PCAP output and text output, respectively:

```
$ MCR LANCP SHOW DEVICE/TRACE EIA/PCAP=TMP.PCAP
$ MCR LANCP SHOW DEVICE/TRACE EIA/PCAP=TMP.PCAP/OUTPUTFILE=TRACE.TXT
```

  To prevent text output, specify **/OUTPUTFILE=NL:**.

- Select particular trace entries:

```
$ MCR LANCP SHOW DEVICE/TRACE=PK EIA
$ MCR LANCP SHOW DEVICE/TRACE=(STATE,UCHANGE) EIA
$ MCR LANCP SHOW DEVICE/TRACE=(MASK=(%x40,0),UCHANGE) EIA
```

Tip

You can use the time difference to analyze the interrupt spacing if interrupt entries are selected. If you select FSTART and FDONE, you can see how much time the driver is spending in its fork process. If you select XISSUE, you can see how fast transmits are being sent out.

Note

For LANCP commands, tracing is paused when data is first requested and resumed on the last response to LANCP. SDA does not pause tracing like this, but you can pause tracing using the following command:

```
$ MCR LANCP SET DEVICE/TRACE=PAUSE EIA
```

You can then review the trace data in SDA. There is no reason to pause the tracing if a changing trace buffer and redundant entries do not disrupt your workflow.

Note

The implementation of tracing may change slightly across OpenVMS releases. Refer to the LANCP online Help to clarify the syntax for a particular release.