NEC

Vector Engine 2.0

Troubleshooting Guide

SX-Aurora TSUBASA

Note on export

This product (including software) might fall into a category of goods (or services) regulated by the Foreign Exchange and Foreign Trade Law of Japan.
If so, when exporting this product, a permission from the Japanese government is required.
If documents or other materials are necessary to apply for the permission, contact the dealer you purchased the product from or an NEC sales representative.

Preface

This document provides steps to determine whether Vector Engine2.0 (hereafter VE) behavior problems you may experience are caused by VE hardware failures.

This document assumes VEOS for using VEs and related software programs have been installed according to the descriptions of *SX-Aurora TSUBASA Installation Guide*.

Contents

Chapter 1.　　What to Do

This chapter describes what to do when a problem related to VE occurs. If the problem is not solved by performing the following actions or determined to be a hardware problem, run hardware problem diagnostics described in Chapter 2.

1.1　A program does not run successfully

1.1.1　Check how to run the program

The way the program was compiled or executed may be causing the problem. See “Chapter 2 Compilation” and “Chapter 3 Program Execution” of *SX-Aurora TSUBASA Program Execution Quick Guide* to check the way to run the program.

1.1.2　Check the software environment

A software program for Aurora may have been set incorrectly. See “Chapter 4 Software Configuration” of *SX-Aurora TSUBASA Installation Guide* and check the settings of software programs.

1.1.3　Check the status of VEs

Run the following command to check the status of VEs. The result of the following example shows the status is normal.

```
$ /opt/nec/ve/bin/vecmd state get
Vector Engine MMM-Command vx.x.xx
Command:
state -N 0 get
--------------------------------------------------------------------------------
VE0 [24:00.0] [ ONLINE              ] Last Modif:yyyy/mm/dd hh:mm:ss
--------------------------------------------------------------------------------
Result: Success
```

If the status of a VE is OFFLINE or MAINTENANCE, the VE cannot run the program. Follow the steps below to change its status to ONLINE.

・When the status is OFFLINE:

```
$ /opt/nec/ve/bin/vecmd state get
Vector Engine MMM-Command vx.x.xx
Command:
state -N 0 get
--------------------------------------------------------------------------------
VE0 [24:00.0] [ OFFLINE                ] Last Modif:yyyy/mm/dd hh:mm:ss
--------------------------------------------------------------------------------
Result: Success

$ /opt/nec/ve/bin/vecmd state set on
Vector Engine MMM-Command vx.x.xx
Command:
state -N 0 set on
--------------------------------------------------------------------------------
VE0 [24:00.0] [ OFFLINE   – ONLINE   ] Last Modif:yyyy/mm/dd hh:mm:ss
--------------------------------------------------------------------------------
Result: Success
```

・When the status is MAINTENANCE:

```
$ /opt/nec/ve/bin/vecmd state get
Vector Engine MMM-Command vx.x.xx
Command:
state -N 0 get
--------------------------------------------------------------------------------
VE0 [24:00.0] [ MAINTENANCE          ] Last Modif:yyyy/mm/dd hh:mm:ss
--------------------------------------------------------------------------------
Result: Success

$ /opt/nec/ve/bin/vecmd reset card
Vector Engine MMM-Command vx.x.xx
Command:
reset -N 0 card
--------------------------------------------------------------------------------
VE0 reset card Success
--------------------------------------------------------------------------------
Result: Success

$ /opt/nec/ve/bin/vecmd state get
Vector Engine MMM-Command vx.x.xx
Command:
state -N 0 get
--------------------------------------------------------------------------------
VE0 [24:00.0] [ ONLINE               ] Last Modif:yyyy/mm/dd hh:mm:ss
--------------------------------------------------------------------------------
Result: Success
```

1.2　A VE is not recognized

If any VE is not recognized, as shown below, check the following. VE1 is not recognized in the following example.

```
$ /opt/nec/ve/bin/vecmd state get
Vector Engine MMM-Command vx.x.xx
Command:
state -N 0,2 get
--------------------------------------------------------------------------------
VE0 [24:00.0] [ ONLINE              ] Last Modif:yyyy/mm/dd hh:mm:ss
VE2 [a0:00.0] [ ONLINE              ] Last Modif:yyyy/mm/dd hh:mm:ss
--------------------------------------------------------------------------------
Result: Success
```

* An example of three VEs mounted on a host server with only two of them being recognized

1.2.1　Check the power cables

When the power cable to be connected to a VE is not connected, the VE is not recognized. Check the power cables of VEs are securely connected.

1.2.2　Check the VE software configuration files

There may be an error in the association of VEs used for VE software programs and PCI bus numbers. Delete the following two files and then restart the host server mounting VEs to see whether the problem is solved.

- /var/opt/nec/ve/veos/ve_nodes.json
- /etc/opt/nec/ve/veos/ve_nodes.json

1.3 UNAVAILABLE is shown for the status of a VE

When UNAVAILABLE is shown for the status of a VE, check the following.

```
$ /opt/nec/ve/bin/vecmd state get
Vector Engine MMM-Command vx.x.xx
Command:
state -N 0 get
--------------------------------------------------------------------------------
VE0 [24:00.0] [ UNAVAILABLE        ] Last Modif:yyyy/mm/dd hh:mm:ss
--------------------------------------------------------------------------------
Result: Success
```

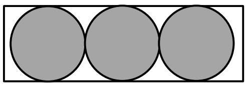
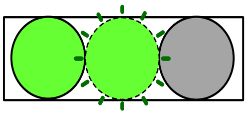
1.3.1 Check the BIOS settings

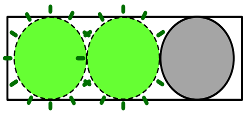
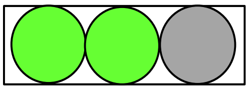
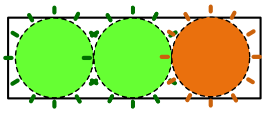
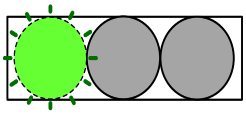
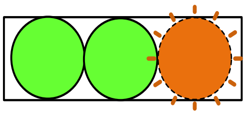
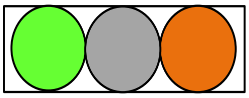
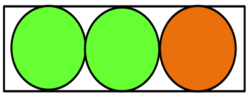
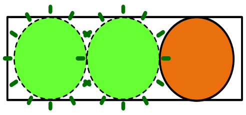
The BIOS settings may be causing the VE problem. Check values of the BIOS are set as follows on the host server mounting the VE.

Item	Value
PCIe Link Speed	Gen3 or larger
PCIe Hot Plug	Disable
ASPM (Active State Power Management)	Disable

1.4 The amber LED of a VE is lit

Three LEDs located at the top of a VE show the VE status according to their lighting pattern. When the amber LED is lit or flashing, there may be a VE problem. Use the following table for troubleshooting.

No.	Status	How LEDs are lit	Lighting pattern	Remarks	Flash interval
1	Initialization		Off/Off/Off	Before VMC start	-
2			On/Flash/Off	Waiting for a request to	1s

				initialize Gen3	
3			Flash/Flash/ Off	Gen3 initialization is ongoing	1s
4	Normal operation		On/On/ off	Operating	-
5	Debug		Flash/Flash/ Flash	For card location presence. The lighting pattern for identifying the debug target	0.2s
6			Flash/Off/ Off	Flash updating	1s
7	Warning operation		On/On/Flash	Environment warning (high temperature)	1s
8	Error		On/Off/On	Environment error detected (temperature, voltage, current)	1s
9			On/On/On	VE ATT0	1s
10			Flash/Flash/ On	Initialization failed	1s

1.4.1 For the lighting pattern 7

The temperature of the VE is high with performance degradation possibility due to throttling. Revise the cooling conditions of the host server, such as by increasing the system fan rotation speed, to use the VE.

1.4.2 For the lighting pattern 8

The temperature of the VE is high or a failure in the power supply to the VE has occurred. If the problem persists even after turning off and then on the power of the host server, there is a possibility of a VE or host server hardware problem. See “Chapter 2 Hardware Problem Diagnostics” and run hardware diagnostics.

＊ The status of the LEDs does not change by reboot.

1.4.3 For the lighting pattern 9

The VE has a hardware problem.

1.4.4 For the lighting pattern 10

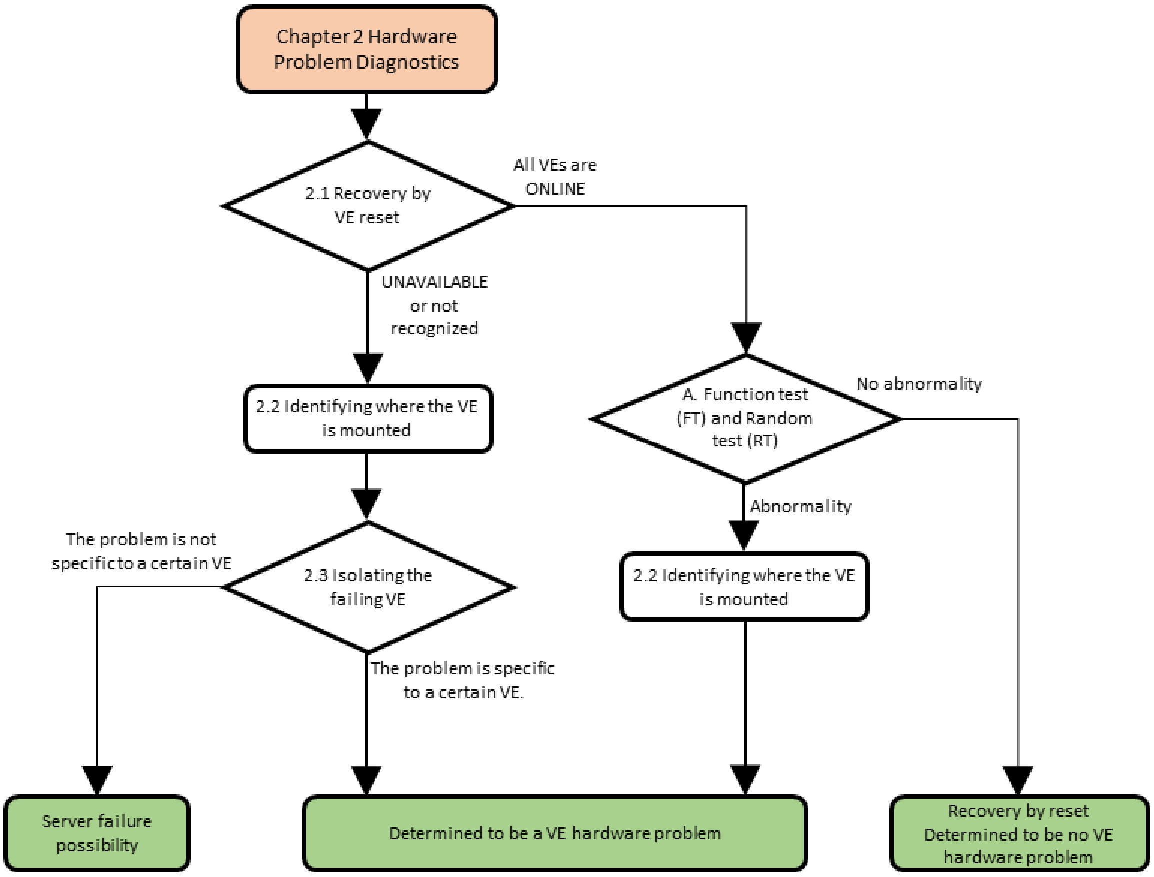
This pattern denotes a VE initialization failure. See “Chapter 2 Hardware Problem Diagnostics” and run hardware diagnostics.

1.5 A program takes longer than usual to run

Throttling may have occurred due to a high temperature of a VE. Revise the cooling conditions, such as by increasing the system fan rotation speed, and see the problem is solved.

Chapter 2. Hardware Problem Diagnostics

Use the flow shown below to identify where the problem occurred.


Chapter 2 Hardware Problem Diagnostics
2.1 Recovery by VE reset
All VEs are ONLINE
UNAVAILABLE or not recognized
2.2 Identifying where the VE is mounted
A. Function test (FT) and Random test (RT)
No abnormality
Abnormality
The problem is not specific to a certain VE
2.3 Isolating the failing VE
2.2 Identifying where the VE is mounted
The problem is specific to a certain VE.
Server failure possibility
Determined to be a VE hardware problem
Recovery by reset Determined to be no VE hardware problem

2.1 Recovery by VE reset

Check whether the problem is solved by resetting VE cards as shown below. If the problem is solved, proceed to (1), if not, proceed to (2).

```
$ /opt/nec/ve/bin/vecmd reset card
Vector Engine MMM-Command vx.x.xx
Command:
reset -N 0 card
--------------------------------------------------------------------------------
VE0 reset card Success
--------------------------------------------------------------------------------
Result: Success

$ /opt/nec/ve/bin/vecmd state get
Vector Engine MMM-Command vx.x.xx
Command:
state -N 0 get
--------------------------------------------------------------------------------
VE0 [24:00.0] [ ONLINE                    ] Last Modif:yyyy/mm/dd hh:mm:ss
--------------------------------------------------------------------------------
Result: Success
```

Proceed to (1) for ONLINE
Proceed to (2) for OFFLINE

(1) If the status becomes ONLINE after the reset, you can check whether VEs are normal through the function test (FT) and the random test (RT). Run the tests to see whether a hardware problem has occurred.
FT and RT have functionality to check whether there is any VE card failure. When a failure is detected, logs for diagnosis are reported. While you can run the tests for each VE, you can also run the tests for up to eight VEs at a time. Use the following table to find how long the tests take.

Abbreviation	Name	Time required for running the test for 1 VE	Time required for running the test for 8 VEs
FT	Function Test	Approx. 15 minutes	Approx. 20 minutes
RT	Random Test	Approx. 20 minutes	Approx. 20 minutes

If the tests do not find a problem, you can determine there is no hardware problem on VEs. If a problem is detected, perform “2.2 Identifying where the VE is mounted” to locate where the failing VE is mounted. For information about how to run the tests, see “Appendix Function Test and Random Test.”

(2) If the problem persists after the reset, perform “2.2 Identifying where the VE is mounted” to locate the failing VE for diagnosis.

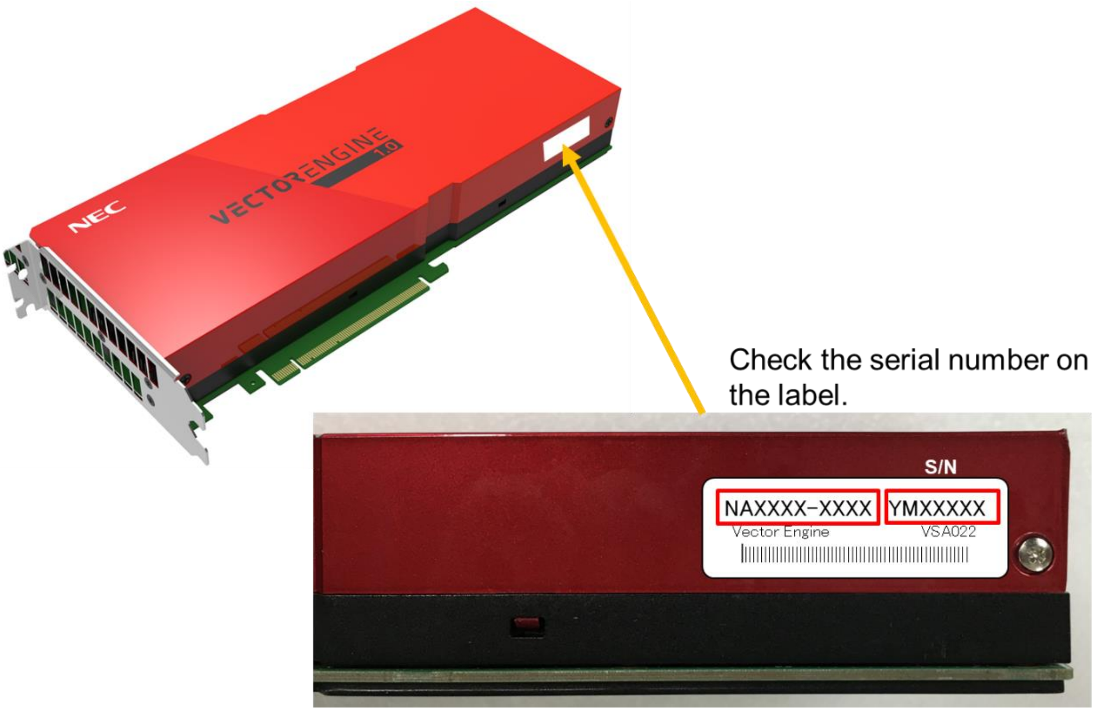
2.2 Identifying where the VE is mounted

(1) Run the following command from the OS to check the serial number of the suspect VE.

```
$ /opt/nec/ve/bin/vecmd info
Vector Engine MMM-Command v1.3.37
Command:
info -N 0,1,2,3,4,5,6,7
--------------------------------------------------------------------------------
Time                                  : 2021/01/21 12:04:50
Attached VEs                          : 8
MMM Version                            : 1.3.37
[VE0 : ALL   ]
VE State                             : UNAVAILABLE
VE Model                              : 1
Product Type                          : 131
Cores                                 : 8
VE Chip S/N                           : 50475747363116000e03020a00000000
VE Board S/N                           : 2BJ480078
                                  .
                                  .
                                  .
-----------------------------------------------------------------------------------
Result: Success
```

Check here

(2) Stop the host server, remove VEs and see their labels at their bottom to check serial numbers. The VE with the serial number checked in (1) is the suspect.



Check the serial number on the label.

2.3　Isolating the failing VE

Mount the VE whose location is identified in “2.2 Identifying where the VE is mounted” in a different PCIe slot and then run the following command again.

```
$ /opt/nec/ve/bin/vecmd state get
Vector Engine MMM-Command vx.x.xx
Command:
state -N 0 get
-------------------------------------------------------------------------------
VE0 [24:00.0] [ ONLINE                ] Last Modif:yyyy/mm/dd hh:mm:ss
-------------------------------------------------------------------------------
Result: Success
```

If the result is the same before and after relocation, a host server failure is possible.

If the result has changed, you can determine the problem is a VE hardware problem.

Appendix A. Function Test (FT) and Random Test (RT)

The following are commands required for FT and RT:

・FT

```
# /opt/nec/ve/bin/vecmd state set off
# /opt/nec/ve/bin/vecmd state set mnt
# /opt/nec/ve/td/ft/all_ft.py
```

・RT

```
# /opt/nec/ve/bin/vecmd state set off
# /opt/nec/ve/bin/vecmd state set mnt
# /opt/nec/ve/bin/vecmd tdctl start rt
```

When no error messages are shown after running the third command, you can determine that VEs do not have a problem. For details of the procedures and message information, see the sections below:

A-1. FT

(1) You can run FT only when the status of VEs is MAINTENANCE. Check the status of VEs before running FT and if the status is other than MAINTENANCE, you need to change the status.

i) Run the following command to check the VE status.

```
$ /opt/nec/ve/bin/vecmd state get
```

When the VE status is "MAINTENANCE,” you can run FT.

If the VE status is ”ONLINE” or ”OFFLINE,” change the status to ”MAINTENANCE.” For ONLINE, perform ii), and for OFFLINE, perform iii).

The following is an example of what is shown on the display when the VE status is checked.

```
$ /opt/nec/ve/bin/vecmd state get
Vector Engine MMM-Command vx.x.xx
Command:
state -N 0,1,2,3,4,5,6,7 get
-------------------------------------------------------------------------------
VE0 [xx:xx.x] [ MAINTENANCE          ] Last Modif:xxxx/xx/xx xx:xx:xx
VE1 [xx:xx.x] [ MAINTENANCE          ] Last Modif:xxxx/xx/xx xx:xx:xx
VE2 [xx:xx.x] [ MAINTENANCE          ] Last Modif:xxxx/xx/xx xx:xx:xx
VE3 [xx:xx.x] [ MAINTENANCE          ] Last Modif:xxxx/xx/xx xx:xx:xx
VE4 [xx:xx.x] [ MAINTENANCE          ] Last Modif:xxxx/xx/xx xx:xx:xx
VE5 [xx:xx.x] [ MAINTENANCE          ] Last Modif:xxxx/xx/xx xx:xx:xx
VE6 [xx:xx.x] [ MAINTENANCE          ] Last Modif:xxxx/xx/xx xx:xx:xx
VE7 [xx:xx.x] [ MAINTENANCE          ] Last Modif:xxxx/xx/xx xx:xx:xx
-------------------------------------------------------------------------------
Result: Success
```

Check here

ii) When the VE status is "ONLINE," change the status to "OFFLINE" first and then to "MAINTENANCE."

Run the following command to change the VE status to “OFFLINE.”

```
$ /opt/nec/ve/bin/vecmd state set off
```

After you run the command, the VE status after the change is shown on the display. Check the status has changed to "OFFLINE."

The following is an example of what is shown on the display when the VE status is updated to "OFFLINE."

```
$ /opt/nec/ve/bin/vecmd state set off
Vector Engine MMM-Command vx.x.xx
```

```
Command:
state -N 0,1,2,3,4,5,6,7 set off
--------------------------------------------------------------------------------
Wait(xx:xx) for VE0[OK] VE1[OK] VE2[OK] VE3[OK] VE4[OK] VE5[OK] VE6[OK]
VE7[OK]

VE0 [xx:xx.x] [ ONLINE            - OFFLINE           ]
VE1 [xx:xx.x] [ ONLINE            - OFFLINE           ]
VE2 [xx:xx.x] [ ONLINE            - OFFLINE           ]
VE3 [xx:xx.x] [ ONLINE            - OFFLINE           ]
VE4 [xx:xx.x] [ ONLINE            - OFFLINE           ]
VE5 [xx:xx.x] [ ONLINE            - OFFLINE           ]
VE6 [xx:xx.x] [ ONLINE            - OFFLINE           ]
VE7 [xx:xx.x] [ ONLINE            - OFFLINE           ]
--------------------------------------------------------------------------------
Result: Success
```

Check here

Run the following command to change the VE status to "MAINTENANCE."

```
$ /opt/nec/ve/bin/vecmd state set mnt
```

After you run the command, the VE status after the change is shown on the display. Check the status has changed to "MAINTENANCE."
To make sure, perform i) to check the VE status.

The following is an example of what is shown on the display when the VE status is updated to "MAINTENANCE."

```
$ /opt/nec/ve/bin/vecmd state set mnt
Vector Engine MMM-Command vx.x.xx
Command:
state -N 0,1,2,3,4,5,6,7 set mnt
--------------------------------------------------------------------------------
Wait(xx:xx) for VE0[OK] VE1[OK] VE2[OK] VE3[OK] VE4[OK] VE5[OK] VE6[OK] VE7[OK]
```

```
VE0 [xx:xx.x] [ OFFLINE              - MAINTENANCE*         ]
VE1 [xx:xx.x] [ OFFLINE              - MAINTENANCE*         ]
VE2 [xx:xx.x] [ OFFLINE              - MAINTENANCE*         ]
VE3 [xx:xx.x] [ OFFLINE              - MAINTENANCE*         ]
VE4 [xx:xx.x] [ OFFLINE              - MAINTENANCE*         ]
VE5 [xx:xx.x] [ OFFLINE              - MAINTENANCE*         ]
VE6 [xx:xx.x] [ OFFLINE              - MAINTENANCE*         ]
VE7 [xx:xx.x] [ OFFLINE              - MAINTENANCE*         ]
--------------------------------------------------------------------------------
Result: Success
```

Check here

iii) When the VE status is ”OFFLINE,” run the following command to change the VE status to "MAINTENANCE."

```
$ /opt/nec/ve/bin/vecmd state set mnt
```

After you run the command, the VE status after the change is shown on the display. Check the status has changed to "MAINTENANCE."

To make sure, perform i) to check the VE status.
The following is an example of what is shown on the display when the VE status is updated to “MAINTENANCE.”

```
$ /opt/nec/ve/bin/vecmd state set mnt
Vector Engine MMM-Command vx.x.xx
Command:
state -N 0,1,2,3,4,5,6,7 set mnt
--------------------------------------------------------------------------------
Wait(xx:xx)  for  VE0[OK]  VE1[OK]  VE2[OK]  VE3[OK]  VE4[OK]  VE5[OK]  VE6[OK]
VE7[OK]

VE0 [xx:xx.x] [ OFFLINE              - MAINTENANCE*         ]
VE1 [xx:xx.x] [ OFFLINE              - MAINTENANCE*         ]
```

```
VE2 [xx:xx.x] [ OFFLINE            - MAINTENANCE*        ]
VE3 [xx:xx.x] [ OFFLINE            - MAINTENANCE*        ]
VE4 [xx:xx.x] [ OFFLINE            - MAINTENANCE*        ]
VE5 [xx:xx.x] [ OFFLINE            - MAINTENANCE*        ]
VE6 [xx:xx.x] [ OFFLINE            - MAINTENANCE*        ]
VE7 [xx:xx.x] [ OFFLINE            - MAINTENANCE*        ]
--------------------------------------------------------------------------------
Result: Success
```

Check here

(2) Run the command.

```
 $ /opt/nec/ve/td/ft/all_ft.py
```

While up to eight VEs can be mounted on a host, this script automatically identifies VEs actually mounted and runs single_VE or multi_VE FT. (VE numbers with no VEs mounted are automatically skipped.) Because Multi_VE FT runs for all combinations, it is committed 28 times when there are eight VEs (singly-linked).

You can specify VEs to be tested by specifying -N.

Examples of specifying -N:

Specifying an individual VE	-N 1	…	Runs only on VE1
Specifying a range of VEs	-N 0-1	…	Runs on VE0 and 1
Specifying VEs by listing	-N 0,1,4	…	Runs on VE0, 1, and 4

You do not need to mind the VE specification order as it is sorted.

Note: If you specify a VE that does not exist, the test fails with the following message.

```
--------------------------------------------------------------------------------
FT-test START failed
VE2 failed to execute "not found VE2"
--------------------------------------------------------------------------------
```

(A) An example of logs for running the command with no options when there are VE0 and VE1:

First, the FT for single-VE runs for VE0 and VE1 concurrently. When this FT completes, the FT for multi-VE, which runs across VE0⇔VE1, runs.

```
% all_ft.py
Vector Engine MMM-Command v1.0.1
Command:
tdctl -N 0 start ft FT.cata
--------------------------------------------------------------------------------
FT-test START 10:10:22 FT.cata
```

```
--------------------------------------------------------------------------------
Result: Success
Vector Engine MMM-Command v1.0.1
Command:
tdctl -N 1 start ft FT.cata
--------------------------------------------------------------------------------
FT-test START 10:10:22 FT.cata
--------------------------------------------------------------------------------
Result: Success
.
Function Test RUNNING : 1% done.
Function Test RUNNING : 1% done.
Function Test RUNNING : 2% done.
Function Test RUNNING : 2% done.
Function Test RUNNING : 3% done.
Function Test RUNNING : 3% done.
Function Test RUNNING : 4% done.
Function Test RUNNING : 4% done.
Function Test RUNNING : 5% done.
Function Test RUNNING : 5% done.
(lines omitted for brevity)
Function Test RUNNING : 99% done.
Function Test Finished.
Vector Engine MMM-Command v1.0.1
Command:
tdctl -N 0,1 start ft FT_2VE.cata
--------------------------------------------------------------------------------
FT-test START 10:19:35 FT_2VE.cata
--------------------------------------------------------------------------------
Result: Success
Function Test RUNNING : 7% done.
Function Test RUNNING : 22% done.
Function Test RUNNING : 27% done.
Function Test RUNNING : 42% done.
Function Test RUNNING : 47% done.
Function Test RUNNING : 61% done.
Function Test RUNNING : 67% done.
Function Test RUNNING : 70% done.
(lines omitted for brevity)
Function Test RUNNING : 96% done.
Function Test RUNNING : 100% done.
Function Test Finished.

******************************
Single VE RESULT:
  VE0: GOOD
  VE1: GOOD
******************************

******************************
Multi VE RESULT:
  VE0-1: GOOD
******************************
```

(B) When only VE1 is specified:

Only single_VE FT for VE1 runs. Because a single VE is specified, multi_VE FT does not run.

```
% all_ft.py –N 1
Vector Engine MMM-Command v1.1.3
Command:
tdctl -N 1 start ft FT.cata
--------------------------------------------------------------------------------
FT-test START 14:03:50 FT.cata
--------------------------------------------------------------------------------
Result: Success
Function Test RUNNING : 1% done.
Function Test RUNNING : 2% done.
Function Test RUNNING : 3% done.
```

```
(lines omitted for brevity)
Function Test RUNNING : 96% done.
Function Test RUNNING : 100% done.
Function Test Finished.

*****************************
Single VE RESULT:
  VE1: GOOD
*****************************
```

When the test starts successfully, "Result:Success" is shown.

When starting the test fails, "Result:Failed" is shown.

For example, running the test when the VE status is "ONLINE."

The following is an example of what is shown on the display when running FT for VE0.

■Successfully starting the test

```
$ /opt/nec/ve/bin/vecmd –N 0 tdctl start ft
Vector Engine MMM-Command vx.x.xx
Command:
tdctl -N 0 start ft
-------------------------------------------------------------------------------
FT-test START xx:xx:xx FT.catat
-------------------------------------------------------------------------------
Result: Success
```

Check here

■Failing to start the test

```
$ /opt/nec/ve/bin/vecmd –N 0 tdctl start ft
Vector Engine MMM-Command vx.x.xx
Command:
tdctl -N 0 start ft
----------------------------------------------------------------------
FT-test START failed
VE0 failed to execute "xxxxxxxxxx"
----------------------------------------------------------------------
Result: Failed
```

Check here

(3) FT is suspended by the following two steps:

Stop the shell for committing consecutively (all_ft.py) by Ctl-C.

Immediately stop ftmon and correct the status by Vecmd tdctl stop.

Example of execution)

```
% all_ft.py
Vector Engine MMM-Command v1.1.3
Command:
tdctl -N 0 start ft FT.cata
----------------------------------------------------------------------
FT-test START 14:19:22 FT.cata
----------------------------------------------------------------------
Result: Success
Vector Engine MMM-Command v1.1.3
Command:
tdctl -N 1 start ft FT.cata
----------------------------------------------------------------------
FT-test START 14:19:22 FT.cata
----------------------------------------------------------------------
Result: Success
```

```
Function Test RUNNING : 1% done.
Function Test RUNNING : 1% done.
Function Test RUNNING : 2% done.
Function Test RUNNING : 2% done.
^CTraceback (most recent call last):

%
FTM: [Info] ftmon was killed by stop signal.
FTM: [Info] ftmon was killed by stop signal.
FT ended in abnormal, please hit return key and then please check logfiles.
FT ended in abnormal, please hit return key and then please check logfiles.

% vecmd tdctl stop
Vector Engine MMM-Command v1.1.3
Command:
tdctl -N 0,1 stop
--------------------------------------------------------------------------------
VE0 FT-test STOP 14:19:35 FT.cata
VE1 FT-test STOP 14:19:35 FT.cata
VE0 FT-test TERM terminated (time out=20s)
VE1 FT-test TERM terminated (time out=20s)
--------------------------------------------------------------------------------
Result: Success

% vecmd tdctl state
Vector Engine MMM-Command v1.1.3
Command:
tdctl -N 0,1 state
--------------------------------------------------------------------------------
     State    Type Elapsed    Catalog
VE0 [ERROR] [FT]   0h 0m10s FT.cata   Confirm FT has stopped
VE1 [ERROR] [FT]   0h 0m10s FT.cata   Confirm FT has stopped
--------------------------------------------------------------------------------
```

(4) FT completes in approximately 15 minutes while the time depends on the number of VEs mounted and the performance of the host. See the information on the display at the end of the test to check the test result. To ensure the test result, see the result file directly or use the VE command as described below:

i) When the check test by the display information shown at the end of the test completes, either of the following messages is shown at the end. GOOD is shown for the normal end and BAD for the abnormal end.
Check the details of the log file when the test ends abnormally.

The following is an example of what is shown on the display when the test ends.

■Normal end

```
$ /opt/nec/ve/bin/vecmd –N 0 tdctl start ft
Vector Engine MMM-Command vx.x.xx
Command:
tdctl -N 0 start ft
-------------------------------------------------------------------------------
FT-test START xx:xx:xx FT.cata
-------------------------------------------------------------------------------
Result: Success
$ FT started
Function Test RUNNING : 2% done.
Function Test RUNNING : 3% done.
Function Test RUNNING : 4% done.
Function Test RUNNING : 5% done.
    :
Function Test RUNNING : 100% done.
Function Test Finished.

*****************************
Single VE RESULT:
  VE0: GOOD
  VE1: GOOD
*****************************
                Check here

*****************************
Multi VE RESULT:
  VE0-1: GOOD
*****************************
```

■Abnormal end

```
$ /opt/nec/ve/bin/vecmd –N 0 tdctl start ft
Vector Engine MMM-Command vx.x.xx
Command:tdctl -N 0 start ft
--------------------------------------------------------------------------------
FT-test START xx:xx:xx FT.cata
--------------------------------------------------------------------------------
Result: Success
Function Test RUNNING : 2% done.
Function Test RUNNING : 3% done.
Function Test RUNNING : 4% done.
Function Test RUNNING : 5% done.

Error! Function Test Finished abnormally.
FT ended in abnormal, please hit return key and then please check logfiles.

******************************
Single VE RESULT:
  VE0: GOOD
  VE1: BAD
******************************
                 Check here
******************************
Multi VE RESULT:
  VE0-1: GOOD
******************************
```

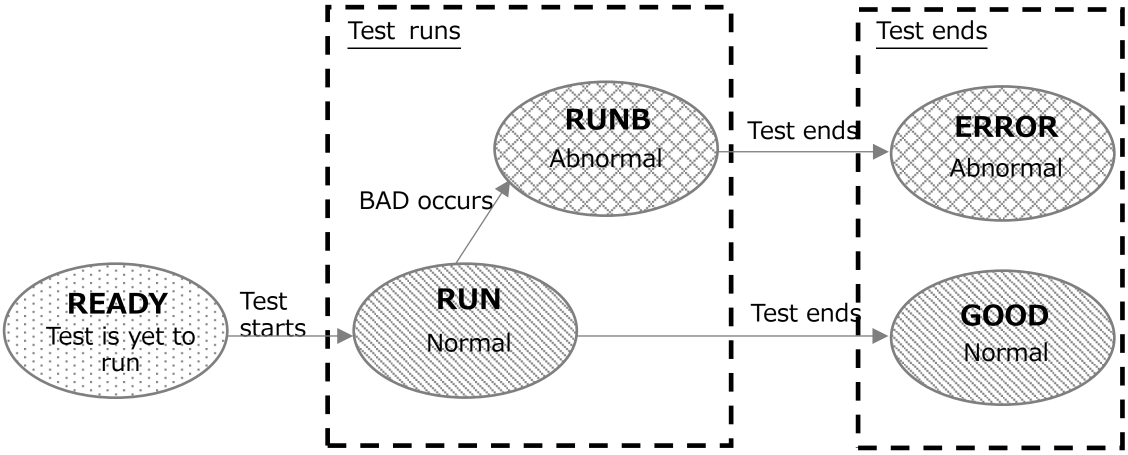
Unless you press the Return key, the result, whether it is normal or abnormal, is not shown at the command prompt. Press the Return key once. (The job itself is running in the background.)

ii) For your information: you can check the test status by running the following VE command.

```
$ /opt/nec/ve/bin/vecmd –N x tdctl state
```

When you run the command, the status of the test is shown in the "State" section next to the VE number for the specified VE.

The following illustration shows test status shown in "State."



The following is an example of what is shown on the display when running the command to check the execution status for all VEs.

■Checking before the test: "READY"

```
$ /opt/nec/ve/bin/vecmd tdctl state
Vector Engine MMM-Command vx.x.xx
Command:
tdctl -N 0,1,2,3,4,5,6,7 state
-------------------------------------------------------------------------------
    State    Type Elapsed    Catalog
VE0 [READY] [   ] --h--m--s --------
VE1 [READY] [   ] --h--m--s --------
VE2 [READY] [   ] --h--m--s --------
VE3 [READY] [   ] --h--m--s --------
VE4 [READY] [   ] --h--m--s --------
VE5 [READY] [   ] --h--m--s --------
VE6 [READY] [   ] --h--m--s --------
VE7 [READY] [   ] --h--m--s --------
-------------------------------------------------------------------------------
Result: Success
```

Check here

■Checking during the test

Normal: "RUN"

```
$ /opt/nec/ve/bin/vecmd tdctl state
Vector Engine MMM-Command vx.x.xx
Command:
tdctl -N 0,1,2,3,4,5,6,7 state
------------------------------------------------------------------------------------
      State    Type Elapsed    Catalog
VE0 [RUN   ] [FT] xxhxxmxxs cata_default
VE1 [RUN   ] [FT] xxhxxmxxs cata_default
VE2 [RUN   ] [FT] xxhxxmxxs cata_default
VE3 [RUN   ] [FT] xxhxxmxxs cata_default
VE4 [RUN   ] [FT] xxhxxmxxs cata_default
VE5 [RUN   ] [FT] xxhxxmxxs cata_default
VE6 [RUN   ] [FT] xxhxxmxxs cata_default
VE7 [RUN   ] [FT] xxhxxmxxs cata_default
------------------------------------------------------------------------------------
Result: Success
```

Check here

Abnormal: "RUNB"

```
$ /opt/nec/ve/bin/vecmd tdctl state
Vector Engine MMM-Command vx.x.xx
Command:
tdctl -N 0,1,2,3,4,5,6,7 state
------------------------------------------------------------------------------------
      State    Type Elapsed    Catalog
VE0 [RUNB ] [FT] xxhxxmxxs cata_default
VE1 [RUNB ] [FT] xxhxxmxxs cata_default
VE2 [RUNB ] [FT] xxhxxmxxs cata_default
VE3 [RUNB ] [FT] xxhxxmxxs cata_default
VE4 [RUNB ] [FT] xxhxxmxxs cata_default
```

```
VE5 [RUNB ] [FT] xxhxxmxxs cata_default
VE6 [RUNB ] [FT] xxhxxmxxs cata_default
VE7 [RUNB ] [FT] xxhxxmxxs cata_default
--------------------------------------------------------------------------------
Result: Success
```

Check here

- ■ Checking after the test

Normal: "GOOD"

```
$ /opt/nec/ve/bin/vecmd tdctl state
Vector Engine MMM-Command vx.x.xx
Command:
tdctl -N 0,1,2,3,4,5,6,7 state
--------------------------------------------------------------------------------
    State   Type Elapsed   Catalog
VE0 [GOOD ] [FT]  0h20m45s cata_default
VE1 [GOOD ] [FT]  0h20m45s cata_default
VE2 [GOOD ] [FT]  0h20m45s cata_default
VE3 [GOOD ] [FT]  0h20m45s cata_default
VE4 [GOOD ] [FT]  0h20m45s cata_default
VE5 [GOOD ] [FT]  0h20m45s cata_default
VE6 [GOOD ] [FT]  0h20m45s cata_default
VE7 [GOOD ] [FT]  0h20m45s cata_default
--------------------------------------------------------------------------------
Result: Success
```

Check here

Abnormal: "ERROR"

```
$ /opt/nec/ve/bin/vecmd tdctl state
Vector Engine MMM-Command vx.x.xx
Command:
tdctl -N 0,1,2,3,4,5,6,7 state
--------------------------------------------------------------------------------
    State   Type Elapsed   Catalog
VE0 [ERROR] [FT]  0h20m45s cata_default
VE1 [ERROR] [FT]  0h20m45s cata_default
VE2 [ERROR] [FT]  0h20m45s cata_default
VE3 [ERROR] [FT]  0h20m45s cata_default
VE4 [ERROR] [FT]  0h20m45s cata_default
```

```
VE5 [ERROR] [FT]   0h20m45s cata_default
VE6 [ERROR] [FT]   0h20m45s cata_default
VE7 [ERROR] [FT]   0h20m45s cata_default
--------------------------------------------------------------------------------
Result: Success
```

Check here

A-2. RT

(1) You can run RT only when the status of VEs is MAINTENANCE. Check the status of VEs before running RT and if the status is other than MAINTENANCE, you need to change the status.

i) Run the following command to check the VE status.

```
$ /opt/nec/ve/bin/vecmd state get
```

When the VE status is "MAINTENANCE,” you can run RT.

If the VE status is “ONLINE” or “OFFLINE,” change the status to “MAINTENANCE.” For ONLINE, perform ii), and for OFFLINE, perform iii).

The following is an example of what is shown on the display when the VE status is checked.

```
$ /opt/nec/ve/bin/vecmd state get
Vector Engine MMM-Command vx.x.xx
Command:
state -N 0,1,2,3,4,5,6,7 get
--------------------------------------------------------------------------------
VE0 [xx:xx.x] [ MAINTENANCE         ] Last Modif:xxxx/xx/xx xx:xx:xx
VE1 [xx:xx.x] [ MAINTENANCE         ] Last Modif:xxxx/xx/xx xx:xx:xx
VE2 [xx:xx.x] [ MAINTENANCE         ] Last Modif:xxxx/xx/xx xx:xx:xx
VE3 [xx:xx.x] [ MAINTENANCE         ] Last Modif:xxxx/xx/xx xx:xx:xx
VE4 [xx:xx.x] [ MAINTENANCE         ] Last Modif:xxxx/xx/xx xx:xx:xx
VE5 [xx:xx.x] [ MAINTENANCE         ] Last Modif:xxxx/xx/xx xx:xx:xx
VE6 [xx:xx.x] [ MAINTENANCE         ] Last Modif:xxxx/xx/xx xx:xx:xx
VE7 [xx:xx.x] [ MAINTENANCE         ] Last Modif:xxxx/xx/xx xx:xx:xx
--------------------------------------------------------------------------------
Result: Success
```

Check here

ii) When the VE status is "ONLINE," change the status to "OFFLINE" first and then to "MAINTENANCE."

Run the following command to change the VE status to “OFFLINE.”

```
$ /opt/nec/ve/bin/vecmd state set off
```

After you run the command, the VE status after the change is shown on the display. Check the status has changed to "OFFLINE."

The following is an example of what is shown on the display when the VE status is updated to “OFFLINE.”

```
$ /opt/nec/ve/bin/vecmd state set off
Vector Engine MMM-Command vx.x.xx
Command:
state -N 0,1,2,3,4,5,6,7 set off
--------------------------------------------------------------------------------
Wait(xx:xx)  for  VE0[OK]  VE1[OK]  VE2[OK]  VE3[OK]  VE4[OK]  VE5[OK]  VE6[OK]
VE7[OK]

VE0 [xx:xx.x] [ ONLINE              - OFFLINE             ]
VE1 [xx:xx.x] [ ONLINE              - OFFLINE             ]
VE2 [xx:xx.x] [ ONLINE              - OFFLINE             ]
VE3 [xx:xx.x] [ ONLINE              - OFFLINE             ]
VE4 [xx:xx.x] [ ONLINE              - OFFLINE             ]
VE5 [xx:xx.x] [ ONLINE              - OFFLINE             ]
VE6 [xx:xx.x] [ ONLINE              - OFFLINE             ]
VE7 [xx:xx.x] [ ONLINE              - OFFLINE             ]
--------------------------------------------------------------------------------
Result: Success
```

Check here

Run the following command to change the VE status to "MAINTENANCE."

```
$ /opt/nec/ve/bin/vecmd state set mnt
```

After you run the command, the VE status after the change is shown on the display. Check the status has changed to "MAINTENANCE."

To make sure, perform i) to check the VE status.

The following is an example of what is shown on the display when the VE status is updated to “MAINTENANCE.”

```
$ /opt/nec/ve/bin/vecmd state set mnt
Vector Engine MMM-Command vx.x.xx
Command:
state -N 0,1,2,3,4,5,6,7 set mnt
--------------------------------------------------------------------------------
Wait(xx:xx) for VE0[OK] VE1[OK] VE2[OK] VE3[OK] VE4[OK] VE5[OK] VE6[OK] VE7[OK]

VE0 [xx:xx.x] [ OFFLINE           - MAINTENANCE*        ]
VE1 [xx:xx.x] [ OFFLINE           - MAINTENANCE*        ]
VE2 [xx:xx.x] [ OFFLINE           - MAINTENANCE*        ]
VE3 [xx:xx.x] [ OFFLINE           - MAINTENANCE*        ]
VE4 [xx:xx.x] [ OFFLINE           - MAINTENANCE*        ]
VE5 [xx:xx.x] [ OFFLINE           - MAINTENANCE*        ]
VE6 [xx:xx.x] [ OFFLINE           - MAINTENANCE*        ]
VE7 [xx:xx.x] [ OFFLINE           - MAINTENANCE*        ]
--------------------------------------------------------------------------------
Result: Success
```

Check here

iii) When the VE status is “OFFLINE,” run the following command to change the VE status to "MAINTENANCE."

```
$ /opt/nec/ve/bin/vecmd state set mnt
```

After you run the command, the VE status after the change is shown. Check the status has changed to "MAINTENANCE."
To make sure, perform i) to check the VE status.

The following is an example of what is shown on the display when the VE status is updated to “MAINTENANCE.”

```
$ /opt/nec/ve/bin/vecmd state set mnt
Vector Engine MMM-Command vx.x.xx
Command:
state -N 0,1,2,3,4,5,6,7 set mnt
--------------------------------------------------------------------------------
Wait(xx:xx)  for  VE0[OK]  VE1[OK]  VE2[OK]  VE3[OK]  VE4[OK]  VE5[OK]  VE6[OK]
VE7[OK]

VE0 [xx:xx.x] [ OFFLINE          - MAINTENANCE*       ]
VE1 [xx:xx.x] [ OFFLINE          - MAINTENANCE*       ]
VE2 [xx:xx.x] [ OFFLINE          - MAINTENANCE*       ]
VE3 [xx:xx.x] [ OFFLINE          - MAINTENANCE*       ]
VE4 [xx:xx.x] [ OFFLINE          - MAINTENANCE*       ]
VE5 [xx:xx.x] [ OFFLINE          - MAINTENANCE*       ]
VE6 [xx:xx.x] [ OFFLINE          - MAINTENANCE*       ]
VE7 [xx:xx.x] [ OFFLINE          - MAINTENANCE*       ]
--------------------------------------------------------------------------------
Result: Success
```

Check here

(2) Run the following VE command to start the test.

```
$ /opt/nec/ve/bin/vecmd –N x tdctl start rt
```

Option to specify target VEs.

・Omit the option when all VEs are targeted.

(Example) `/opt/nec/ve/bin/vecmd tdctl start rt`

・Use commas to specify your target VEs

(Example) `/opt/nec/ve/bin/vecmd-N 0,2,4,6 tdctl start rt`

・Use a hyphen to specify target VEs that are consecutive

(Example) `/opt/nec/ve/bin/vecmd -N 0-3 tdctl start rt`

When the test starts successfully, “Result:Success” is shown.

When starting the test fails, “Result:Failed” is shown.

For example, running the test when the VE status is "ONLINE."

The following is an example of what is shown on the display when running RT for all VEs.

■Successfully starting the test

```
$ /opt/nec/ve/bin/vecmd tdctl start rt
Vector Engine MMM-Command vx.x.xx
Command:
tdctl -N 0,1,2,3,4,5,6,7 start rt
--------------------------------------------------------------------------
RT-test START xx:xx:xx cata_default
--------------------------------------------------------------------------
Result: Success
[veadmin@a1sbm_000 ~]$ RT started
```

Check here

■Failing to start the test

```
$ /opt/nec/ve/bin/vecmd tdctl start rt
Vector Engine MMM-Command vx.x.xx
Command:
tdctl -N 0,1,2,3,4,5,6,7 start rt
------------------------------------------------------------------------------
RT-test START failed
VE0 failed to execute "xxxxxxxxxx"
------------------------------------------------------------------------------
Result: Failed
```

Check here

(3) Check the RT result. When the condition for completing the test (elapsing 20 minutes) is satisfied, the test automatically completes.

To check the test result, you can see information on your display at the end of the test or use the VE command as described below:

i) When the check test by the display information shown at the end of the test completes, either of the following messages is shown.

When the test ends successfully, you see “RT ended in normal, please hit return key.”

When the test ends abnormally, you see “RT ended in abnormal, please hit return key and then please check logfiles.”

The following is an example of what is shown on the display when the test ends.

■Normal end

```
$ /opt/nec/ve/bin/vecmd tdctl start rt
Vector Engine MMM-Command vx.x.xx
Command:
tdctl -N 0,1,2,3,4,5,6,7 start rt
----------------------------------------------------------------------------
RT-test START xx:xx:xx cata_default
----------------------------------------------------------------------------
Result: Success
$ RT started
test_executing. -- case=xxxxxxxx
test_executing. -- case=xxxxxxxx
test_executing. -- case=xxxxxxxx
test_executing. -- case=xxxxxxxx
test_executing. -- case=xxxxxxxx
              .
              .
              .
              .

RT ended in normal, please hit return key.
```

Check here

■Abnormal end

```
$ /opt/nec/ve/bin/vecmd tdctl start rt
Vector Engine MMM-Command vx.x.xx
Command:
tdctl -N 0,1,2,3,4,5,6,7 start rt
-----------------------------------------------------------------------------
RT-test START xx:xx:xx cata_default
-----------------------------------------------------------------------------
Result: Success
$ RT started
test_executing. -- case=xxxxxxxx
test_executing. -- case=xxxxxxxx
test_executing. -- case=xxxxxxxx
test_executing. -- case=xxxxxxxx
test_executing. -- case=xxxxxxxx
                    .
                    .
                    .
                    .

RT ended in abnormal, please hit return key and then please check logfiles.
```

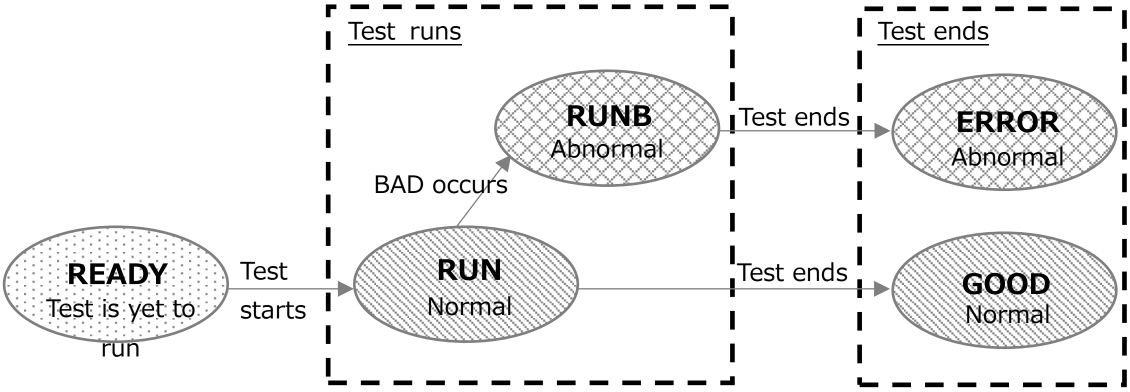
Check here

ii)　You can check how the test has run by the following VE command.

```
$ /opt/nec/ve/bin/vecmd –N x tdctl state
```

When you run the command, the status of the test is shown in the "State" section next to the VE number for the specified VE.

The following illustration shows test status shown in "State."



The following is an example of what is shown on the display when running the command to check the execution status for all VEs.

■Checking before the test: “READY”

```
$ /opt/nec/ve/bin/vecmd tdctl state
Vector Engine MMM-Command vx.x.xx
Command:
tdctl -N 0,1,2,3,4,5,6,7 state
--------------------------------------------------------------------------------
     State    Type Elapsed    Catalog
VE0 [READY] [   ] --h--m--s --------
VE1 [READY] [   ] --h--m--s --------
VE2 [READY] [   ] --h--m--s --------
VE3 [READY] [   ] --h--m--s --------
VE4 [READY] [   ] --h--m--s --------
VE5 [READY] [   ] --h--m--s --------
VE6 [READY] [   ] --h--m--s --------
VE7 [READY] [   ] --h--m--s --------
--------------------------------------------------------------------------------
Result: Success
```

Check here

■Checking during the test

Normal: “RUN”

```
$ /opt/nec/ve/bin/vecmd tdctl state
Vector Engine MMM-Command vx.x.xx
Command:
tdctl -N 0,1,2,3,4,5,6,7 state
--------------------------------------------------------------------------
     State    Type Elapsed    Catalog
VE0 [RUN   ] [RT] xxhxxmxxs cata_default
VE1 [RUN   ] [RT] xxhxxmxxs cata_default
VE2 [RUN   ] [RT] xxhxxmxxs cata_default
VE3 [RUN   ] [RT] xxhxxmxxs cata_default
VE4 [RUN   ] [RT] xxhxxmxxs cata_default
VE5 [RUN   ] [RT] xxhxxmxxs cata_default
VE6 [RUN   ] [RT] xxhxxmxxs cata_default
VE7 [RUN   ] [RT] xxhxxmxxs cata_default
--------------------------------------------------------------------------
Result: Success
```

Check here

Abnormal: “RUNB”

```
$ /opt/nec/ve/bin/vecmd tdctl state
Vector Engine MMM-Command vx.x.xx
Command:
tdctl -N 0,1,2,3,4,5,6,7 state
--------------------------------------------------------------------------
     State    Type Elapsed    Catalog
VE0 [RUNB ] [RT] xxhxxmxxs cata_default
VE1 [RUNB ] [RT] xxhxxmxxs cata_default
VE2 [RUNB ] [RT] xxhxxmxxs cata_default
VE3 [RUNB ] [RT] xxhxxmxxs cata_default
VE4 [RUNB ] [RT] xxhxxmxxs cata_default
```

```
VE5 [RUNB ] [RT] xxhxxmxxs cata_default
VE6 [RUNB ] [RT] xxhxxmxxs cata_default
VE7 [RUNB ] [RT] xxhxxmxxs cata_default
------------------------------------------------------------------------------
Result: Success
```

Check here

Checking after the test

Normal: “GOOD”

```
$ /opt/nec/ve/bin/vecmd tdctl state
Vector Engine MMM-Command vx.x.xx
Command:
tdctl -N 0,1,2,3,4,5,6,7 state
------------------------------------------------------------------------------
     State    Type Elapsed    Catalog
VE0 [GOOD ] [RT]   0h20m45s cata_default
VE1 [GOOD ] [RT]   0h20m45s cata_default
VE2 [GOOD ] [RT]   0h20m45s cata_default
VE3 [GOOD ] [RT]   0h20m45s cata_default
VE4 [GOOD ] [RT]   0h20m45s cata_default
VE5 [GOOD ] [RT]   0h20m45s cata_default
VE6 [GOOD ] [RT]   0h20m45s cata_default
VE7 [GOOD ] [RT]   0h20m45s cata_default
------------------------------------------------------------------------------
Result: Success
```

Check here

Abnormal: “ERROR”

```
$ /opt/nec/ve/bin/vecmd tdctl state
Vector Engine MMM-Command vx.x.xx
Command:
tdctl -N 0,1,2,3,4,5,6,7 state
--------------------------------------------------------------------------------
    State    Type Elapsed    Catalog
VE0 [ERROR] [RT]   0h20m45s cata_default
VE1 [ERROR] [RT]   0h20m45s cata_default
VE2 [ERROR] [RT]   0h20m45s cata_default
VE3 [ERROR] [RT]   0h20m45s cata_default
VE4 [ERROR] [RT]   0h20m45s cata_default
VE5 [ERROR] [RT]   0h20m45s cata_default
VE6 [ERROR] [RT]   0h20m45s cata_default
VE7 [ERROR] [RT]   0h20m45s cata_default
--------------------------------------------------------------------------------
Result: Success
```

Check here

(4) When the condition for completing the test (elapsing 20 minutes) is satisfied, RT automatically completes.
If you want to end RT before 20 minutes elapses, run an end command.
There are two types of end commands; the stop command and the kill command.
The following shows how to run the commands:

i) The stop command (to end RT before completion at a case): run the following command to stop ongoing RT before completion.
Note that elapsing three minutes or more after running the command forcefully terminates the test and no statistical information is reported.

```
$ /opt/nec/ve/bin/vecmd –N x tdctl stop
```

The following is an example of what is shown on the display when running the command above for all VEs.

```
$ /opt/nec/ve/bin/vecmd tdctl stop
Vector Engine MMM-Command vx.x.xx
Command:
tdctl -N 0,1,2,3,4,5,6,7 stop
---------------------------------------------------------------------------
VE0 RT-test STOP xx:xx:xx cata_default
VE1 RT-test STOP xx:xx:xx cata_default
VE2 RT-test STOP xx:xx:xx cata_default
VE3 RT-test STOP xx:xx:xx cata_default
VE4 RT-test STOP xx:xx:xx cata_default
VE5 RT-test STOP xx:xx:xx cata_default
VE6 RT-test STOP xx:xx:xx cata_default
VE7 RT-test STOP xx:xx:xx cata_default
---------------------------------------------------------------------------
Result: Success
$ RT ended in normal, please hit return key.
```

Note: If you have started the test by specifying target VEs by –N, specify the same VEs (as specified at the start of the test).

If you omit the VE specification by –N, the test stops for all VEs.

The following is an example of what is shown on the display for forced termination for passing three minutes or more after running the command.

```
$ /opt/nec/ve/bin/vecmd tdctl stop
Vector Engine MMM-Command vx.x.xx
Command:
tdctl -N 0,1,2,3,4,5,6,7 stop
--------------------------------------------------------------------------------
VE0 RT-test TERM xx:xx:xx cata_default
VE1 RT-test TERM xx:xx:xx cata_default
VE2 RT-test TERM xx:xx:xx cata_default
VE3 RT-test TERM xx:xx:xx cata_default
VE4 RT-test TERM xx:xx:xx cata_default
VE5 RT-test TERM xx:xx:xx cata_default
VE6 RT-test TERM xx:xx:xx cata_default
VE7 RT-test TERM xx:xx:xx cata_default
--------------------------------------------------------------------------------
Result: Success
$ RT ended in abnormal, please hit return key and then please check logfiles.
```

ii) The kill command (to forcefully terminate RT immediately): run the following command to forcefully terminate ongoing RT.
No statistical information is reported when RT is forcefully terminated.

```
$ /opt/nec/ve/bin/vecmd –N x tdctl kill
```

The following is an example of what is shown on the display when running the command above for all VEs.

```
$ /opt/nec/ve/bin/vecmd tdctl kill
Vector Engine MMM-Command vx.x.xx
```

```
Command:
tdctl -N 0,1,2,3,4,5,6,7 kill
--------------------------------------------------------------------------------
VE0 RT-test TERM xx:xx:xx cata_default
VE1 RT-test TERM xx:xx:xx cata_default
VE2 RT-test TERM xx:xx:xx cata_default
VE3 RT-test TERM xx:xx:xx cata_default
VE4 RT-test TERM xx:xx:xx cata_default
VE5 RT-test TERM xx:xx:xx cata_default
VE6 RT-test TERM xx:xx:xx cata_default
VE7 RT-test TERM xx:xx:xx cata_default
--------------------------------------------------------------------------------
Result: Success
$ RT ended in abnormal, please hit return key and then please check logfiles.
```

Note1: Depending on the timing to run the command, the message, “mv: ‘/var/opt/nec/mmm/td/xxxxxxxx’ can’t stat:There’s no such file or directory” may be shown.

Note 2: If you have started the test by specifying VEs by –N, specify the same VEs (as specified at the start of the test).
If you omit the VE specification by –N, the test stops for all VEs.

Appendix B.　Publishing History

B-1. Publishing History

2022	March	First edition

SX-Aurora TSUBASA System Software

Vector Engine 2.0

Troubleshooting Guide

2022 March First Edition

NEC Corporation

7-1, Shiba 5-chome
Minato-ku, Tokyo
Phone: +81-3-3454-1111