

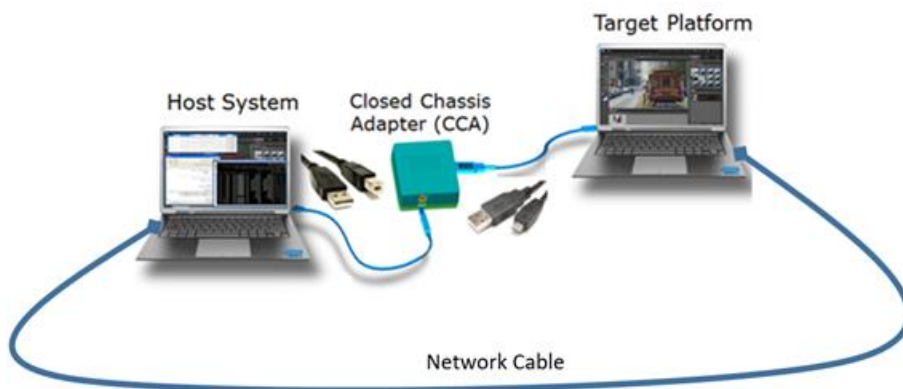
Non-NPK ETW capture

Version: 0.1

Introduction:

The Intel® Trace Hub (Intel® TH) hardware is a set of functional blocks with the ability to perform full system debugging. That is, the Intel Trace Hub is not intended for hardware only or software only, but for full system debug. Specifically, it allows for testing the interaction of hardware and software as they produce complex system behaviors.

Traditional NPK system setup.

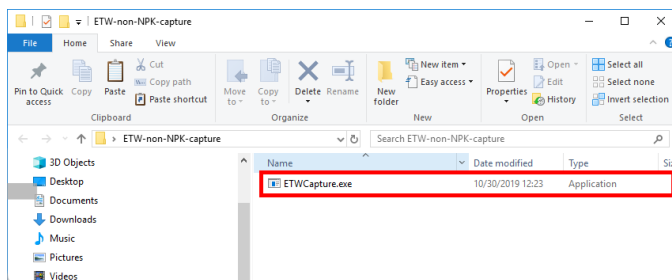


Problem Statement:

Though NPK provides full-fledged solution to capture software (ETW) Traces , Hard ware and firmware traces , Still the solution requires a complex hardware setup even to capture windows ETW traces. It would be ideal to have stand alone application which allows to enable , capture ETW traces and create the trace file with PVT decodable format without using the traditional NPK setup.

Stand alone Application (ETWCapture.exe):

To overcome hardware dependency, Standalone application is implemented which has capability to enable ETW providers , and start/stop tracing and to be able to create the trace file which can be decoded with PVT decodable format.



Follow the instructions below to capture ETW traces with standalone application.

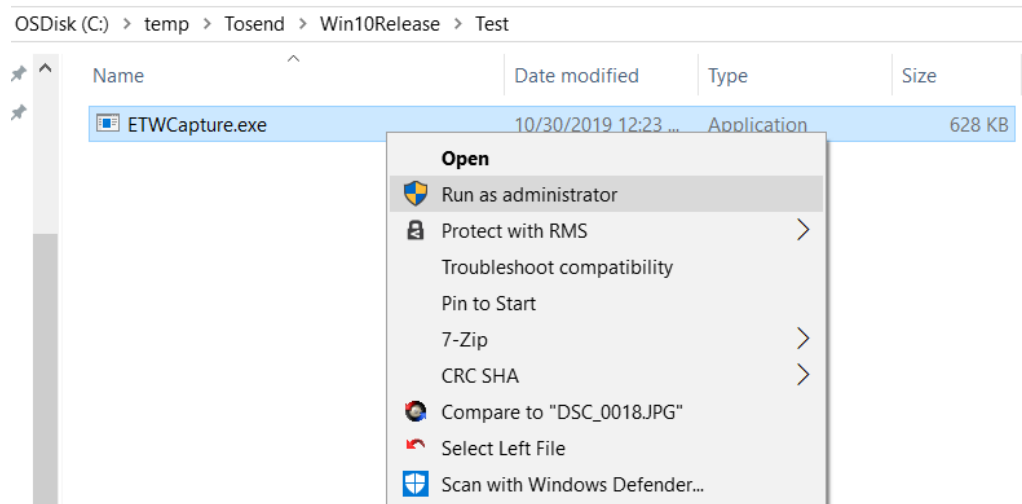
Launch Application

The application should be running in **"Administrator"** mode to be able to capture traces.

1. Launch From Explorer

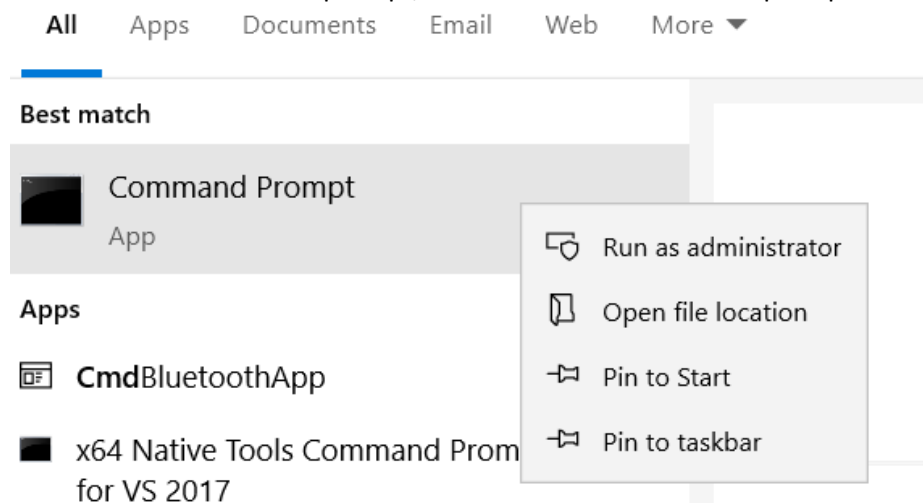
Locate the folder where "ETWCapture.exe" is copied /exist.

From Explorer window → Right Click → Run as administrator.



2. Launch From command prompt

To launch from command prompt, make sure to run command prompt in admin mode.



Traverse to the directory where "ETWCapture.exe" is copied /exist.

>>./ETWCapture.exe // Execute in command line

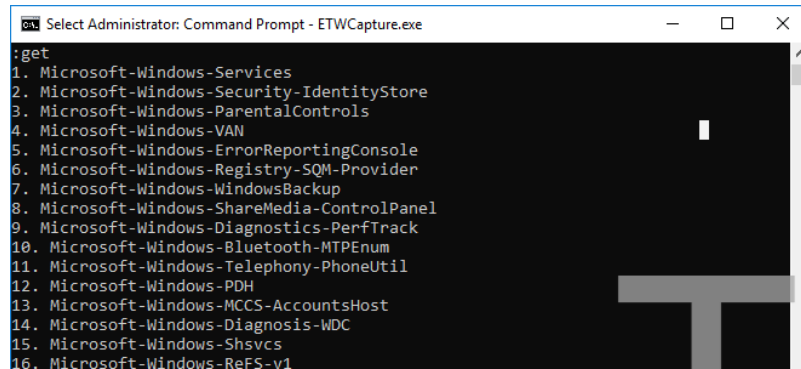
Application command line options

Application provides command line options to be able to enable ETW providers , start capturing traces , stop ..etc.

List of options provided in command line.

➤ **get**

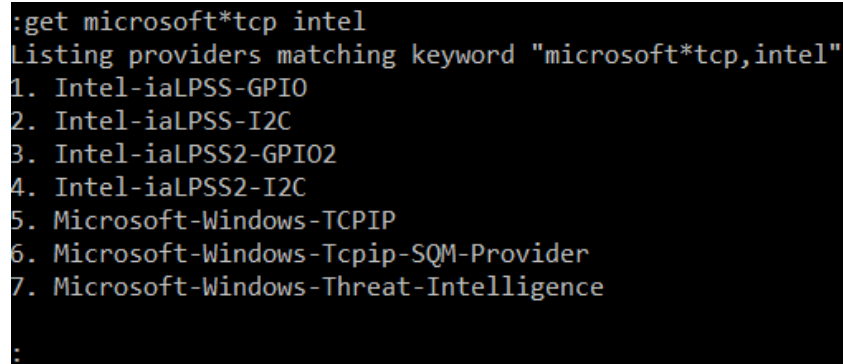
- Prints all slist of Providers available with the system
- Note : Application will display only the providers registered with Windows Operating system.



```
Select Administrator: Command Prompt - ETWCapture.exe
: get
1. Microsoft-Windows-Services
2. Microsoft-Windows-Security-IdentityStore
3. Microsoft-Windows-ParentalControls
4. Microsoft-Windows-VAN
5. Microsoft-Windows-ErrorReportingConsole
6. Microsoft-Windows-Registry-SQM-Provider
7. Microsoft-Windows-WindowsBackup
8. Microsoft-Windows-ShareMedia-ControlPanel
9. Microsoft-Windows-Diagnostics-PerfTrack
10. Microsoft-Windows-Bluetooth-MTPEnum
11. Microsoft-Windows-Telephony-PhoneUtil
12. Microsoft-Windows-PDH
13. Microsoft-Windows-MCCS-AccountsHost
14. Microsoft-Windows-Diagnosis-WDC
15. Microsoft-Windows-Shsvcs
16. Microsoft-Windows-ReFS-v1
```

➤ **get <pattern-1> <pattern-2> <pattern-3><pattern-n>**

- Prints List of Providers matching the patterns
- A pattern may be a partial name , or the sequence of keywords with wild card. Example of pattern below
 - `get microsoft*tcp intel`
 - This command lists the providers matching the provider name containing microsoft*tcp , intel keyword.



```
:get microsoft*tcp intel
Listing providers matching keyword "microsoft*tcp,intel"
1. Intel-iaLPSS-GPIO
2. Intel-iaLPSS-I2C
3. Intel-iaLPSS2-GPIO2
4. Intel-iaLPSS2-I2C
5. Microsoft-Windows-TCPIP
6. Microsoft-Windows-Tcpip-SQM-Provider
7. Microsoft-Windows-Threat-Intelligence
:
```

- Note : Application will display only the providers registered with Windows Operating system.

➤ **tracelevel**

- Prints the current default trace level set.

➤ **Tracelevel <level>**

- Sets the default trace level.
- 1:CRITICAL, 2:ERROR , 3: WARNING , 4: INFO , 5:VERBOSE
- If tracelevel is not set by default “5:VERBOSE” is set.ss

➤ **enable <providername-1> <providername-2> <providername-3><providername-n>**

- Enable list of ETW providers for tshe trace
- <Provider name> : The provider name should be matching exactly as it has been listed with the 'get' option.

When the command is successful the message will be prompted with successful

```
:enable Intel-iaLPSS-GPIO Microsoft-Windows-Tcpip-SQM-Provider Intel-iaLPSS-I2C
EnableProviders successful
```

If any of the providers failed to be enabled then appropriate error message will be shown with provider name as shown below

```
:enable Intel-iaLPSS-GPIO Microsoft-Windows-Tcpip-SQM-Provider Intel-iaLPSS-I2C Intel-ERROR
Enable providers Failed for few Selected providers
Review the Errors below. Check for exact name for providers and retry . View NPK-Capture-Bin.txt file for more info
Intel-ERROR, Failed with error code : 87
```

➤ **enable**

<providername1> [optional]<tracelevel> | [optional]<matchanykeyword> | [optional]<matchall keyword> <providername2> <providername3>.....

- Enables providers with individual options
- <Tracelevel> - optional : Trace level can be one of following values as given below
 - 1:CRITICAL, 2:ERROR, 3: WARNING, 4: INFO, 5:VERBOSE
 - IF none of the tracelevel options provided then default level set from <tracelevel> command will be used. Refer tracelevel command for more info.s
- <matchanykeyword>: optional : The mask value to filter specific keyword (in number only). By default matchanykeyword is set to 0 if its not provided.
- <smatchallkeyword >: optional: The mask value to filter all keyword which matches(in number only). By default smatchallkeyword is set to 0 if its not provided.
- Sample for enabling providers with individual optionsss

```
:enable Intel-iaLPSS-GPIO|2|255|111 Microsoft-Windows-Tcpip-SQM-Provider|3|1 I
ntel-iaLPSS-I2C|1|0|9
EnableProviders successful
```

- Note: All above options are optional , these options can be set for each provider when enabling multiple providers if specific tracelevel and other options need to be set individually.

➤ **disable <providername1> <providername2> <providername3>.....**

- Disable providers . The providers should be already enabled in order to disable providers successfully , otherwise it will fail.

➤ **Start**

- start Trace , which will start capturing real time traces from ETW.
- The output filename also will be printed when the trace is successful. The file name will be generated uniquely and stored in the same application directory.

```
:start
StartTrace successful
Do not close the application
To stop press Enter any key..
```

- Make sure to not to exit the application to continue capturing live traces. If the application is stopped or closed Capturing will be halted and there will not be any traces captured in the file.
- To stop the live traces press “Enter” key . This will create “ETWTRACE-*.bin” file in “C:\ProgramData\ETW-Traces” directory. Pick up the trace file generated with the latest modified date in the directory.

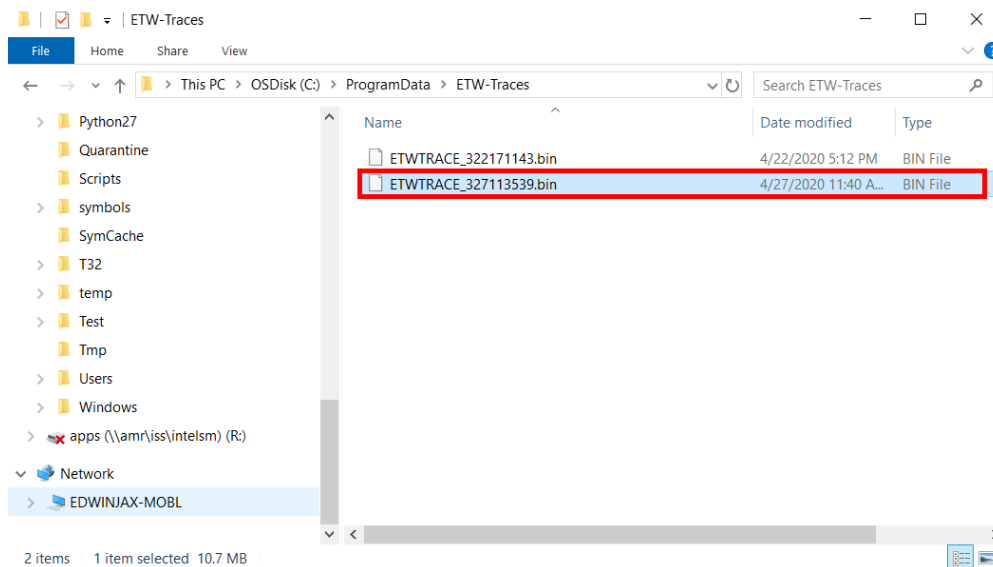
➤ q/stop/exit

- To exit application type ‘q’ or ‘stop’ or ‘exit’ and press enter

Note: The application should be running for the whole session to capture tracing for the providers enabled. If the application exits or terminated all ETW tracing configuration is lost , and it need to be done for each new session.

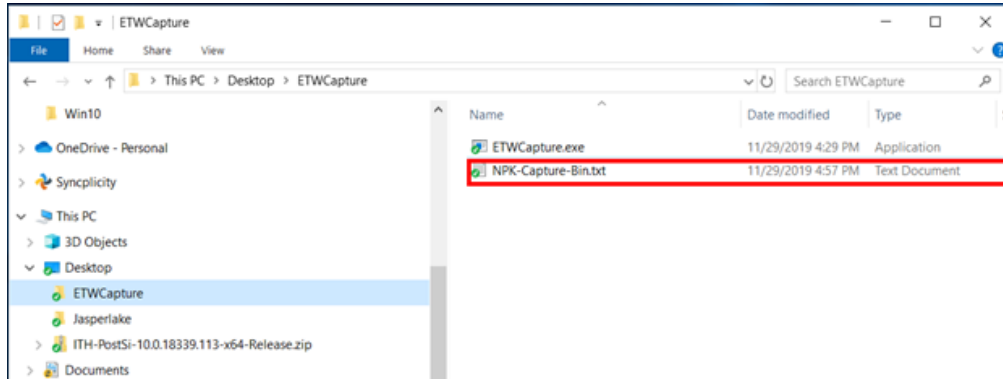
Trace Files

The Trace file names are auto generated starting with ETWTRACE_<timestamp>.bin file for each trace session. The captured traces will be stored in “ETWTRACE-*.bin” file in “C:\ProgramData\ETW-Traces” directory. Pick up the trace file generated with the latest modified date in the directory. Later the file can be imported from PVT for decoding.



Log file

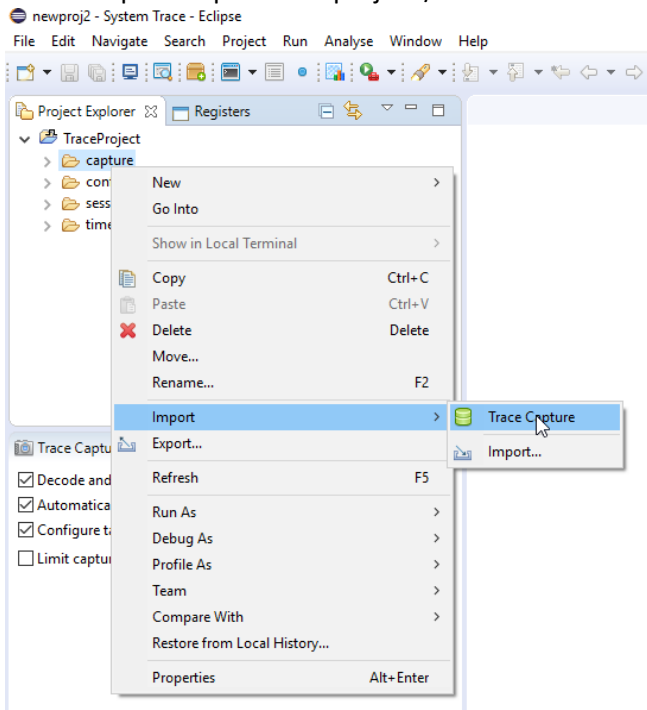
There's a log file "NPK-Capture-bin.txt" created in the same application directory to view the detailed log and error information.



Decode from Eclipse:

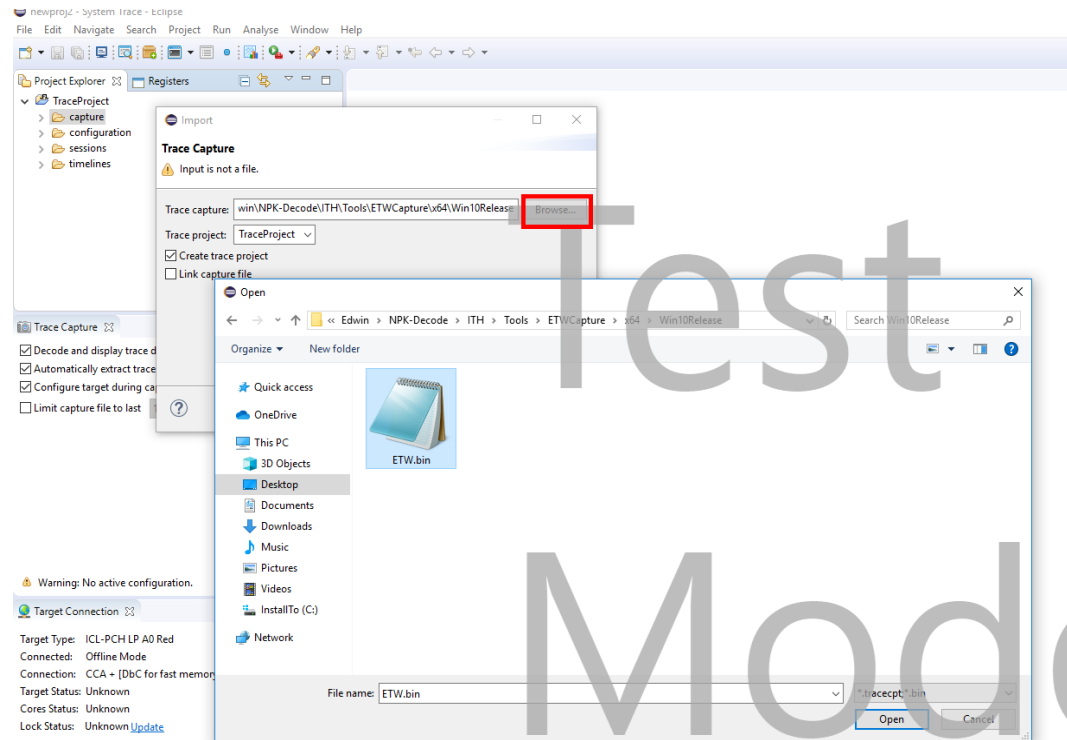
1. Import file to eclipse

a. Open Eclipse Trace project / Create new trace project.



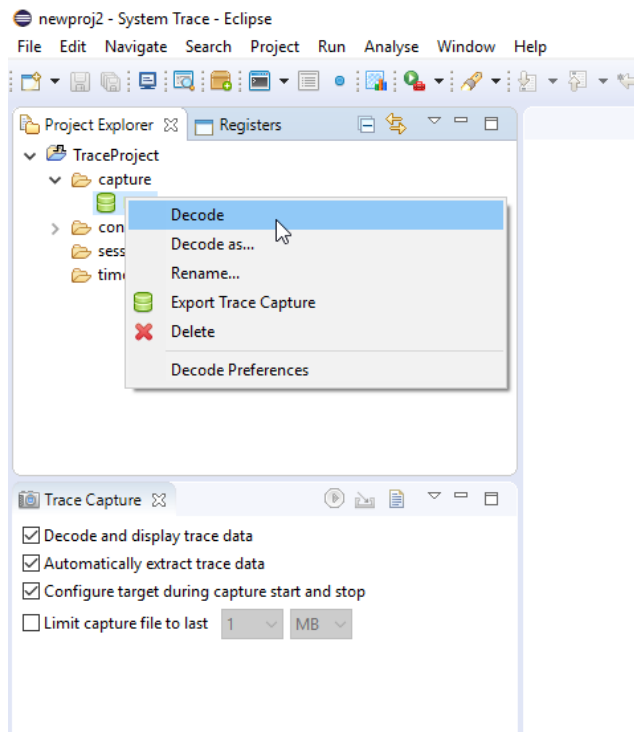
b. Right click on "Capture" in "project Explorer" , → Import→ Trace Capture

c. Click on "Browse" button and locate the "ETWTRACE*.bin" file which is captured.



d. The Imported trace file will be shown under capture directory in project explorer window

2. Decode:



Right click on imported trace file under “Capture” directory and click on decode to decode the traces.

Sample decoded Traces shown below.

newproj - System Trace - MessageView001 - Eclipse

File Edit Navigate Search Project Run Analyse Window Help

Project Explorer: TraceProject, capture, configuration, sessions, ETW_2019_10_30_164735_15, timelines

MessageView001: Synchronize, Auto-Scroll, Type filter text, Filter Off, Search Off

Time	Source	Summary	MasterID
6	Windows	Microsoft-Windows-TCP/IP-->TCP: connection 0x77778D99A235F010 (local=127.0.0.1:51158 remote=127.0.0.1:51157) exists. State = EstablishedState . PID ...	0x0100
7	Windows	Microsoft-Windows-TCP/IP-->TCP: connection 0x77778D99A0B1AE00 (local=10.223.138.29:7680 remote=10.66.249.230:64929) exists. State = TimeWaitState	0x0100
8	Windows	Microsoft-Windows-TCP/IP-->TCP: connection 0x77778D99A0BC6400 (local=10.223.138.29:50702 remote=10.224.224.80:912) exists. State = CloseWaitState	0x0100
9	Windows	Microsoft-Windows-TCP/IP-->TCP: connection 0x77778D99A1EF3420 (local=10.223.138.29:51172 remote=10.66.245.170:7680) exists. State = EstablishedState	0x0100
10	Windows	Microsoft-Windows-TCP/IP-->TCP: connection 0x77778D99A05B2C00 (local=10.223.138.29:49554 remote=40.90.189.152:443) exists. State = EstablishedState	0x0100
11	Windows	Microsoft-Windows-TCP/IP-->TCP: connection 0x77778D99A9B97010 (local=127.0.0.1:51152 remote=127.0.0.1:51159) exists. State = EstablishedState . PID ...	0x0100
12	Windows	Microsoft-Windows-TCP/IP-->TCP: connection 0x77778D99A2BD6C00 (local=10.223.138.29:7680 remote=10.66.126.112:59934) exists. State = EstablishedState	0x0100
13	Windows	Microsoft-Windows-TCP/IP-->TCP: connection 0x77778D99A0F1E010 (local=127.0.0.1:51156 remote=127.0.0.1:51155) exists. State = EstablishedState . PID ...	0x0100
14	Windows	Microsoft-Windows-TCP/IP-->TCP: connection 0x77778D99A0392010 (local=10.223.138.29:51110 remote=10.223.244.30:7680) exists. State = EstablishedState	0x0100
15	Windows	Microsoft-Windows-TCP/IP-->TCP: connection 0x77778D99A0D0C010 (local=127.0.0.1:51157 remote=127.0.0.1:51159) exists. State = EstablishedState . PID ...	0x0100
16	Windows	Microsoft-Windows-TCP/IP-->IP: Interface rundown: Interface = 1, Compartment = 1, Metric = 75, Connected = TRUE, Linkspeed = 0 bps, PhysicalMediumType = ...	0x0100
17	Windows	Microsoft-Windows-TCP/IP-->IP: Interface rundown: Interface = 3, Compartment = 1, Metric = 5, Connected = FALSE, Linkspeed = 18446744073709551615 b... ..	0x0100
18	Windows	Microsoft-Windows-TCP/IP-->IP: Interface rundown: Interface = 19, Compartment = 1, Metric = 35, Connected = TRUE, Linkspeed = 1000000000 bps, Physic... ..	0x0100
19	Windows	Microsoft-Windows-TCP/IP-->IP: Interface rundown: Interface = 1, Compartment = 1, Metric = 75, Connected = TRUE, Linkspeed = 0 bps, PhysicalMediumType = ...	0x0100
20	Windows	Microsoft-Windows-TCP/IP-->IP: Interface rundown: Interface = 3, Compartment = 1, Metric = 5, Connected = FALSE, Linkspeed = 18446744073709551615 b... ..	0x0100
21	Windows	Microsoft-Windows-TCP/IP-->IP: Interface rundown: Interface = 19, Compartment = 1, Metric = 35, Connected = TRUE, Linkspeed = 1000000000 bps, Physic... ..	0x0100

Microsoft-Windows-TCP/IP-->IP: Interface rundown: Interface = 19, Compartment = 1, Metric = 35, Connected = TRUE, Linkspeed = 1000000000 bps, PhysicalMediumType = NdisPhysicalMedium802_3, NetworkCategory = Public, InternetConnectivityStatus = NonInternetConnectivity, Flags = 0x1C2E6102880, Address = 0.0.0.0 (ignore IPv4 address), IPv6 address = f...

Warning: No active configuration.

Target Connection: Target Type: ICL-PCH LP A0 Red, Connected: Offline Mode, Connection: CCA - [D&C for fast memory extraction], Target Status: Unknown, Cores Status: Unknown, Lock Status: Unknown Update

Console: Message Distribution, Programming, Signal Properties

System Debug: 16:39:48 [INFO] Using System Trace installation from "C:\Intel\TraceHub\". 16:47:35 [INFO] Import in progress.

Import completed with 915 Messages.

Type here to search

4:48 PM 10/30/2019