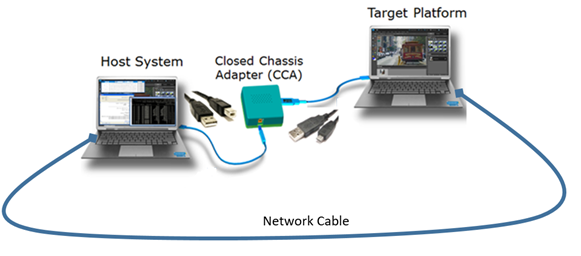
# Non-NPK ETW capture

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.

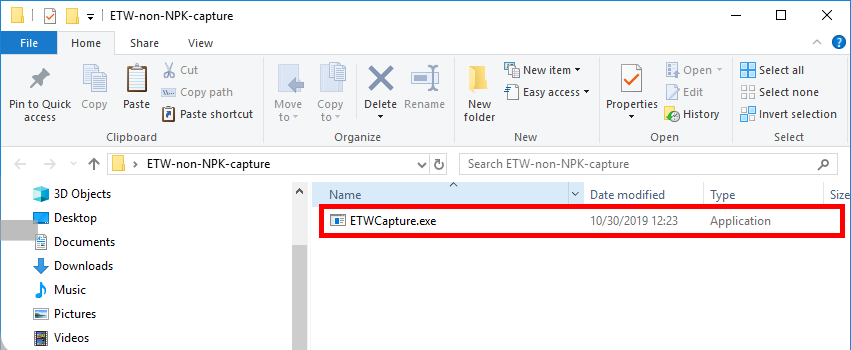


## **Problem Statement:**

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

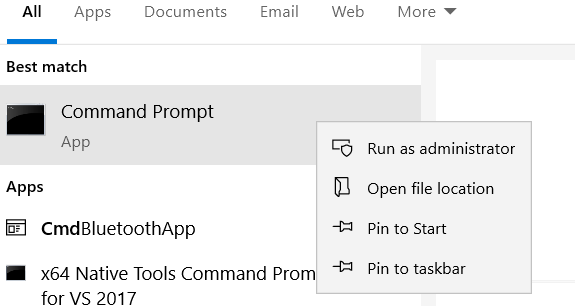
Locate the folder where “ETWCapture.exe” is copied /exist.

From Explorer window 🡪 Right Click 🡪 Run as administrator.

## 

1. 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.

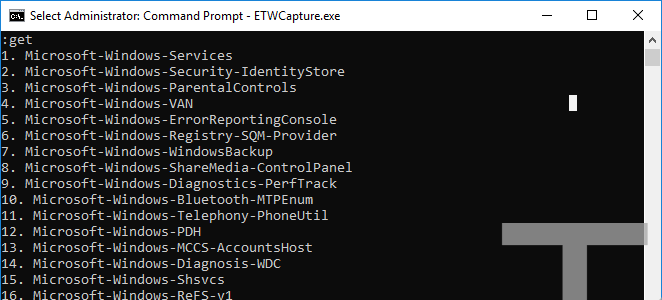
>>./ETWCapture.exe // Execute in command line

## **Application command line options**

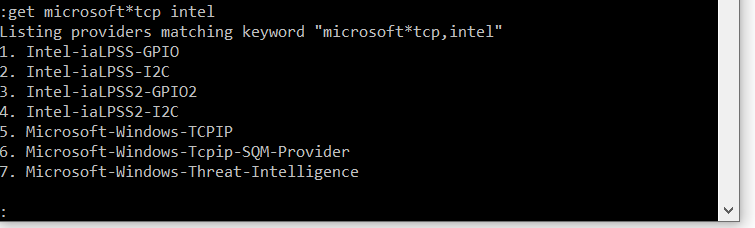
Application provides command line options to be able to enable ETW providers , start capturing traces , stop ..etc.

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.

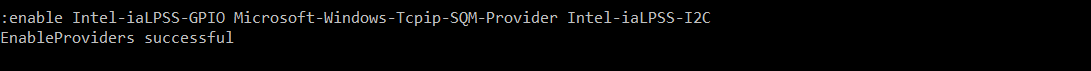
****

* **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 oy 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.

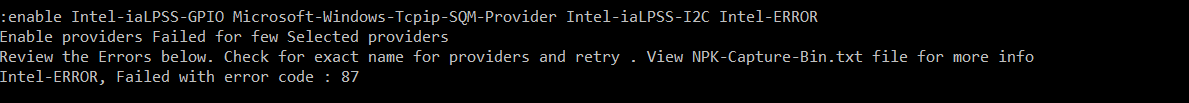


* + Note : Application will display only the providers registered with Windows Operating system.
* **tracelevel**
  + Prints the current default trace level set.
* **Tracelevel <level>**
  + Sets the default trace level.
  + 1:CRITICAL, 2:ERROR , 3: WARNING , 4: INFO , 5:VERBOSE
  + If tracelevel is 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.

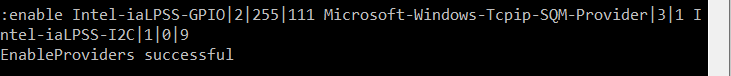
When the command is successful the message will be prompted with successful



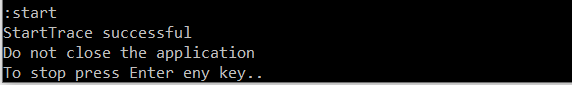
If any of the providers failed to be enabled then appropriate error message will be shown with provider name as shown below



* **enable <providername1>|[optional]<tracelevel>|[optional]<matchanykeyword>|[optional]<matchallkeyword> <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 tracelelevel 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



* + - 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.

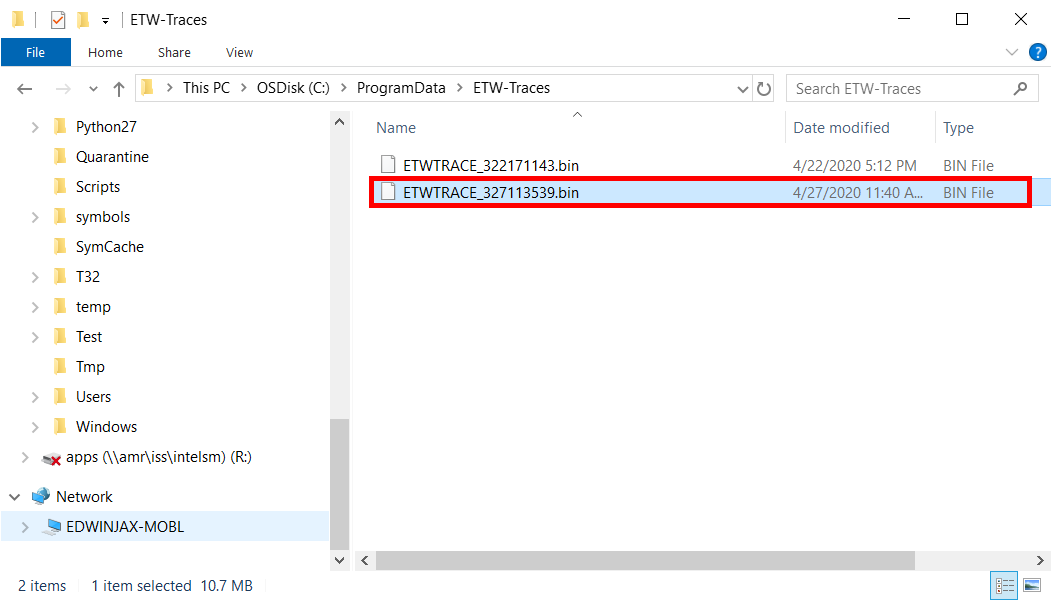


* + Make sure to not to exit the application to continue capturing live traces. If the application is stopped or closed Capturing will be halted and there will not be any traces captured in the file.
  + To stop the live traces press “Enter” key . This will create “ETWTRACE-\*.bin” file in “C:\ProgamData\ETW-Traces” 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.

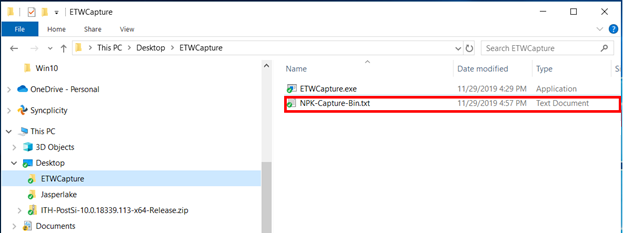
## **Trace Files**

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:\ProgamData\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.



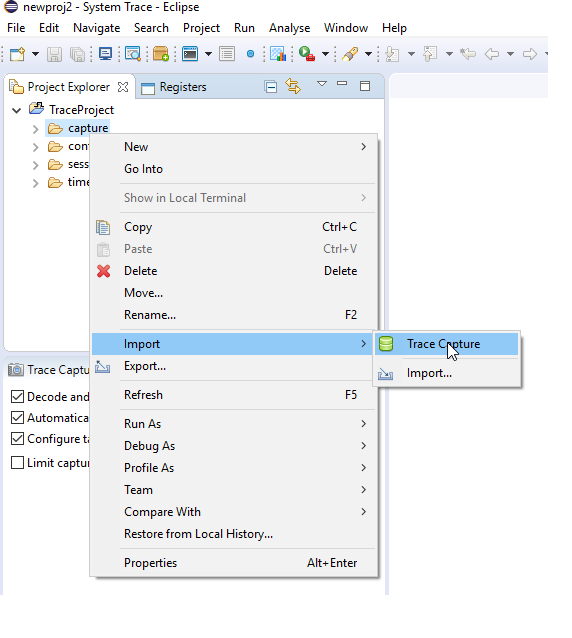
## **Log file**

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

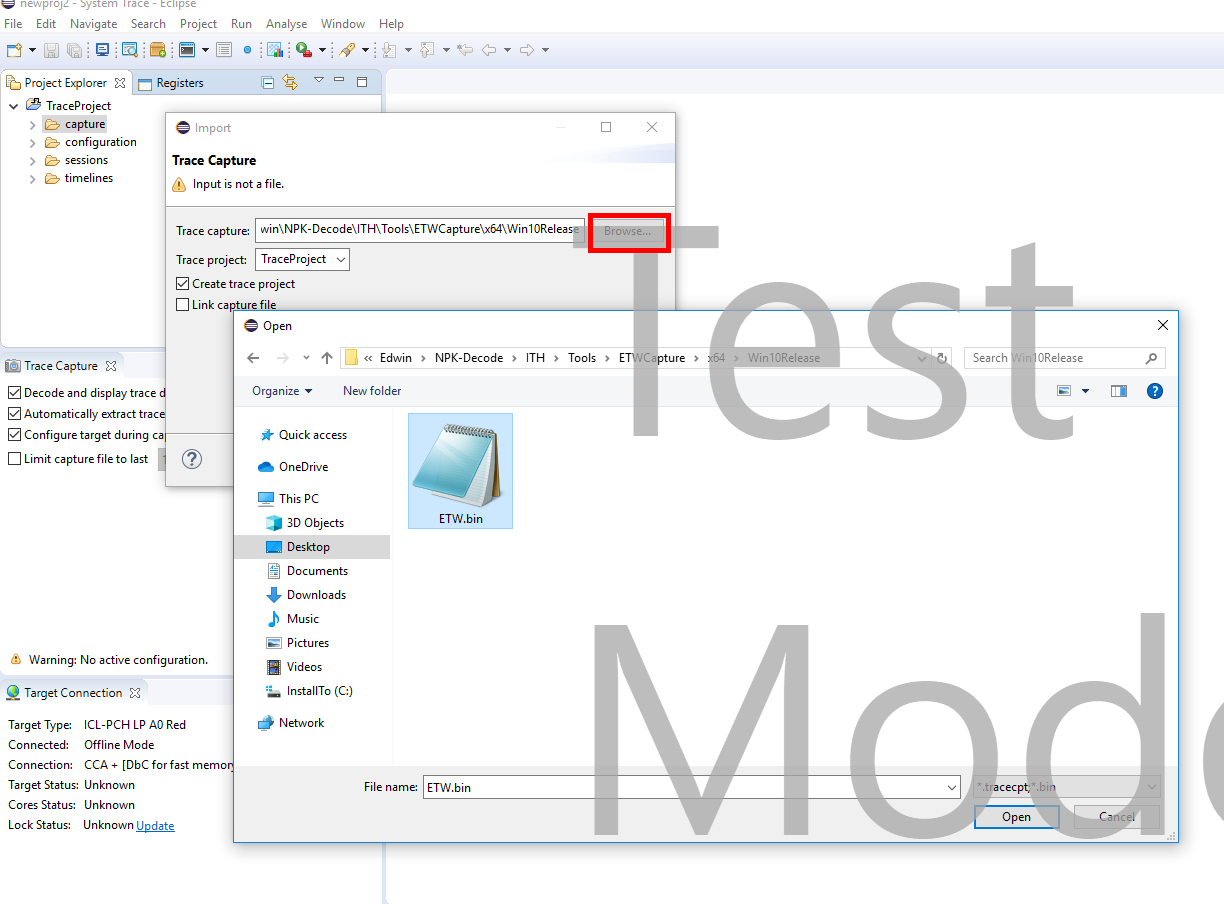


## **Decode from Eclipse:**

1. **Import file to eclipse**
   1. Open Eclipse Trace project / Create new trace project.

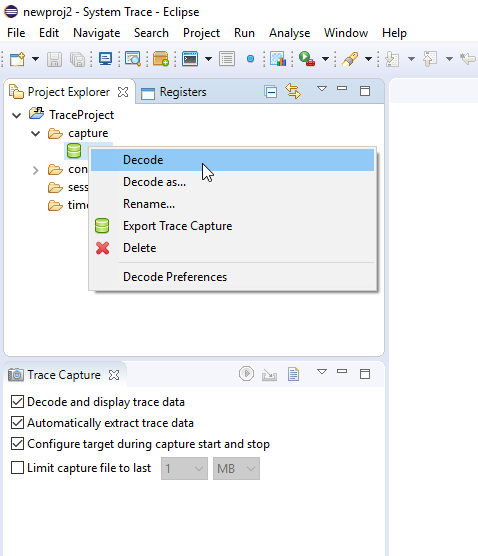
****

* 1. Right click on “Capture” in “project Explorer” , 🡪 Import🡪 Trace Capture
  2. Click on “Browse” button and locate the “ETWTRACE\*.bin” file which is captured.

****

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

1. **Decode**:



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

Sample decoded Traces shown below.

