Intel® Integrated Sensor Solution Trace Tools- User Guide

Legal Notices and Disclaimers

Trace Config Tool:

Purpose:

Get or set firmware trace configurations.

Usage for Windows 8.1/10 Desktop 32/64bit version:

- "TraceConfigTool.exe -getconfig [header] [-ConnectionType <HECI|SimulatedHECI>]": prints the current FW trace configuration or the trace log header if [header] is specified.
- "TraceConfigTool.exe -setconfig <configFile> [volatile] [-ConnectionType <HECI|SimulatedHECI>]": sets the trace configuration according to an xml file. See below for the file format.

Usage for Yocto Linux (GLIBCXX_3.4.21, Kernel 4.1.27) 64bit version:

- TraceConfigTool -getconfig [header]
- TraceConfigTool -setconfig <configFile> [volatile]

Details:

configFile: The path to the configuration file. This file should be in the following xml format:

```
<myConfiguration>
        <TraceInterfacesMask>...</TraceInterfacesMask>
        <TraceModeMask>...</TraceModeMask>
        <BufferSize>...</BufferSize>
        <UARTPort>…</UARTPort>
        <UARTBaudrate>…</UARTBaudrate>
        <I2CBUS>...</I2CBUS>
        <I2CAddress>...</I2CAddress>
        <HostDebugEnabled>...</ HostDebugEnabled>
        <LowestPriority>...</LowestPriority>
        <CompFilterMask>...</CompFilterMask>
</myConfiguration>
```

The xml tags contain the following values:

- ***TraceInterfacesMask***: 1 or more of: HECI (bit 0), I2C (bit 1), LPK (bit 2), UART (bit 3). This should be entered as the decimal equivalent of bitwise OR; for instance, to activate all four, enter "15" in this tag.
- ***TraceModeMask***: buffer disable (0), non-cyclic buffered mode (bit 0), cyclic buffered mode (bit 0+bit 1), block thread (bit 3- if msg queue is full, thread calling syslog() is blocked until queue can accept trace msg). As above, this should be entered as the decimal equivalent of bitwise OR.
- ***BufferSize***: for buffered mode, in KB.
- ***UartPort***: UART port number for trace messages output.

- ***UartBaudrate***: UART Baudrate for trace message output, support 9600 19200 38400 57600 115200 921600 2000000 3000000.
- ***I2CBUS***: The I2C bus on which the FTDI is connected, in the format "0x00".
- ***I2CAddress***: The FTDI slave address, in the format "0x00".
- ***HostDebugEnabled***: (0=disable, 1=enable).
- ***LowestPriority***: 0=emergency, 1=alert, 2=critical, 3=error, 4=warning, 5=notice, 6=info, 7=debug.
- ***CompFilterMask***: A 256-bit mask, expressed as a hexadecimal string, indicating which components will have their trace outputs displayed. A list of up to 16 major components and the up to 16 subcomponents found in each one can be found in the firmware KIT under FW/trace/SI/TraceComponents.xml.
  The major components are not directly represented in the mask; each of their subcomponents are directly activated through a single bit bit. The first 16 bits represent the (up to) 16 subcomponents of the first major component, while the next 16 bits represent the (up to) 16 subcomponents of the second major component, etc. If a major component has fewer than 16 subcomponents, the leftover bits are ignored but must still be present as placeholders; the same is true for leftover groups of 16 bits if there are fewer than 16 major components.

An example for configFile:

```
<?xml version="1.0" encoding="UTF-8"?>
<myConfiguration>
        <TraceInterfacesMask>3</TraceInterfacesMask>
        <TraceModeMask>0</TraceModeMask>
        <BufferSize>2</BufferSize>
        <I2CBUS>0</I2CBUS>
        <I2CAddress>0x5e</I2CAddress>
        <UARTPort>1</UARTPort>
        <UARTBaudrate>115200</UARTBaudrate>
        <HostDebugEnabled>1</HostDebugEnabled>
        <LowestPriority>7</LowestPriority>
        <CompFilterMask>ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff</CompFi
lterMask>
</myConfiguration>
```

Volatile: if volatile, the configuration won't be saved in the firmware after the Integrated Sensor Solution is reset.

Header: if header is specified with –getconfig command, it would get the trace log header structure and print the content on screen.

-ConnectionType <HECI|SimulatedHECI>: The connection interface to work with. Use -HECI to connect to physical firmware running on the local computer, or -SimulatedHECI to connect to simulated firmware in the Simics Firmware Only or Galileo execution environments.

Intel® ISS Trace Collector:

Purpose:

Collect trace messages from firmware according to the current configuration and decode/ print the traces to several interfaces. ISSTraceCollector uses UTC time as default. The TraceConfigTool can run and change configuration even while the ISSTraceCollector is running.

Usage for Windows 8.1/10 Desktop 32/64bit version:

ISSTraceCollector <HECI|simulatedHECI> [continuous] [flush] [saveRaw] <catalogPath> <compPath> <logPath>

ISSTraceCollector [decodeRaw] <catalogPath> <compPath> <binPath>

ISSTraceCollector <I2C_COM|I2C_TCP> [I2CPort] [saveRaw] <catalogPath> <compPath> <logPath>

ISSTraceCollector [UART_COM] [COMPort_Baudrate] [saveRaw] <catalogPath> <compPath> <logPath>

Examples:

- ISSTraceCollector I2C_COM COM3 TraceCatalog.xml TraceComponents.xml log
- ISSTraceCollector UART_COM COM3_115200 TraceCatalog.xml TraceComponents.xml log
- ISSTraceCollector HECI continuous TraceCatalog.xml TraceComponents.xml log
- ISSTraceCollector decodeRaw TraceCatalog.xml TraceComponents.xml log.bin

Usage for Yocto Linux (GLIBCXX_3.4.21, Kernel 4.1.27) 64bit version:

ISSTraceCollector HECI [continuous] [flush] <catalogPath> <compPath> <logPath>

Examples:

ISSTraceCollector HECI continuous TraceCatalog.xml TraceComponents.xml log.txt

Details:

Trace interface type: Can be “I2C_COM|UART_COM” for an FTDI port, “I2C_TCP” for Simics, “HECI” for firmware on the local computer, or “simulatedHECI” simulated firmware in the Simics Firmware Only environment.

I2CPort: The name of COM port, or the TCP socket number on which the message will be received.

COMPort_Baudrate: The name of COM port on which the messages will be received, and the baudrate configured to show message. The baudrate need to be aligned with the setting in trace configuration file, default value is 115200 if not specified.

continuous: for continuous tracing. If not continuous, flash the buffer one time and stop.

flush: force sending flush request. In buffer mode, FW would send trace message after get flush request instead of sending the message only when trace buffer full.

saveRaw: save trace log header and raw trace message into <logPath>.bin file.

decodeRaw: decode the trace log header and raw trace message file <logPath> which is saved by [saveRaw] option.

catalogPath and compPath: The path to the two files TraceCatalog.xml and TraceComponent.xml, which are generated along with the firmware image. They can be found in the **FPGA** and the **SI** subfolders of **\FW\trace**.

logPath: The file in which the decoded trace messages should be stored after the ISSTraceCollector tool runs. If [saveRaw] option is selected, trace log header and raw trace message would be saved into [logPath].bin file at the same time.

The trace collector's logging function is compatible with Microsoft's ETW Trace Listener. Run the logman service from a command prompt to begin logging; see the MSDN pages on the trace listener for details. The provider's GUID is {B90CE0B8-84DD-498E-A3BE-40D9DB96414E}.