Hardware Check

Carlos Abido
Core Team

Hard check is tool for validating the hardware configuration. For projects that are already in IPU with no hardware changes, it can help to quickly validate the project's（POR）configuration.

Hardware Check is python script that compares the BoardInfo output data.json file with manually configurable yaml file per projects.

Features:

- Automatically selects between Python 2 and Python 3
- Creates two fail files with different extensions: .html and .log on local host and copy .log to OD Collect during a failure in automation.
- There are two options available to run the hardware check: manually or through the loaders Xwrapper or Satellite. Additionally, two flags are available for automation: 'stop-on-error' and 'continue-on-error'. These flags allow the automation to either stop or continue in case of an error. A log is created for further investigation.
- Automatically selects the correct yaml per siliconfamily and units.

BoardInfo

Board Info is a complex python collection scripts that reports the inventory of various installed components of a given system.

Example: DIMM Installation, CPU SN#, CPU Step, PCIe card , etc

A data.json file is created. It contain formatted data from the most recent run.

BoardInfo: C:\SVSHARE\BoardInfo\BoardInfo.py
data.json: C:\SVSHARE\BoardInfo\Output\data.json

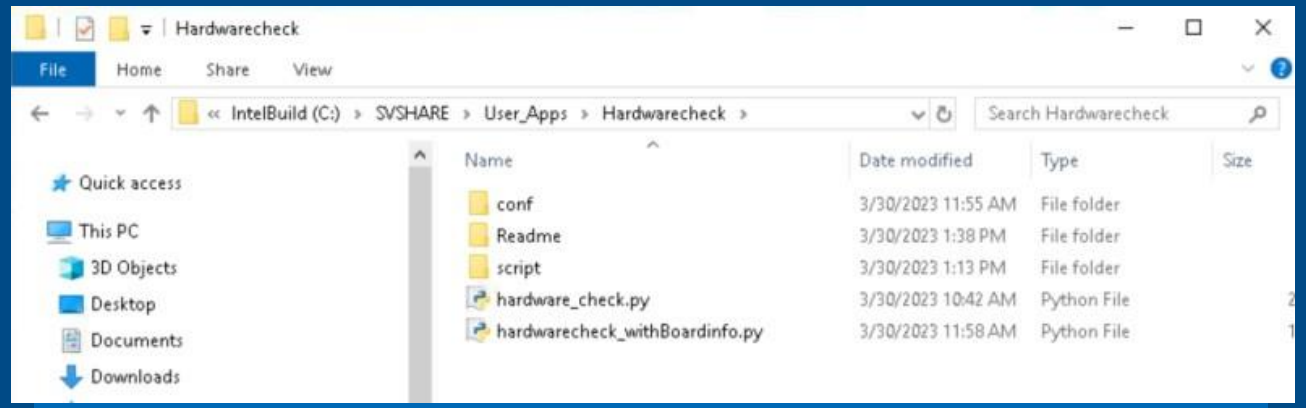
Installation

The hardware_check package is available in Packman.

To install, use the following command:

packman install --name hardware_check

Hardware Check will be installed in:
C:\SVSHARE\User_Apps\Hardwarecheck.



The following folders are included:

- **conf**: contains all yaml configuration files
- **Readme**: This PDF
- **script**: contains Python 2 and Python 3 scripts
- **hardware_check.py**: calls the appropriate script depending on the Python version. Xwrapper or Satellite.
- **hardwarecheck_withBoardinfo.py**: This script calls boardinfo.py –intrusive and hardware_check.py. This script doesn't call the loaders Xwrapper or Satellite. It's call BoardInfo and hardware_check.py with no flag.

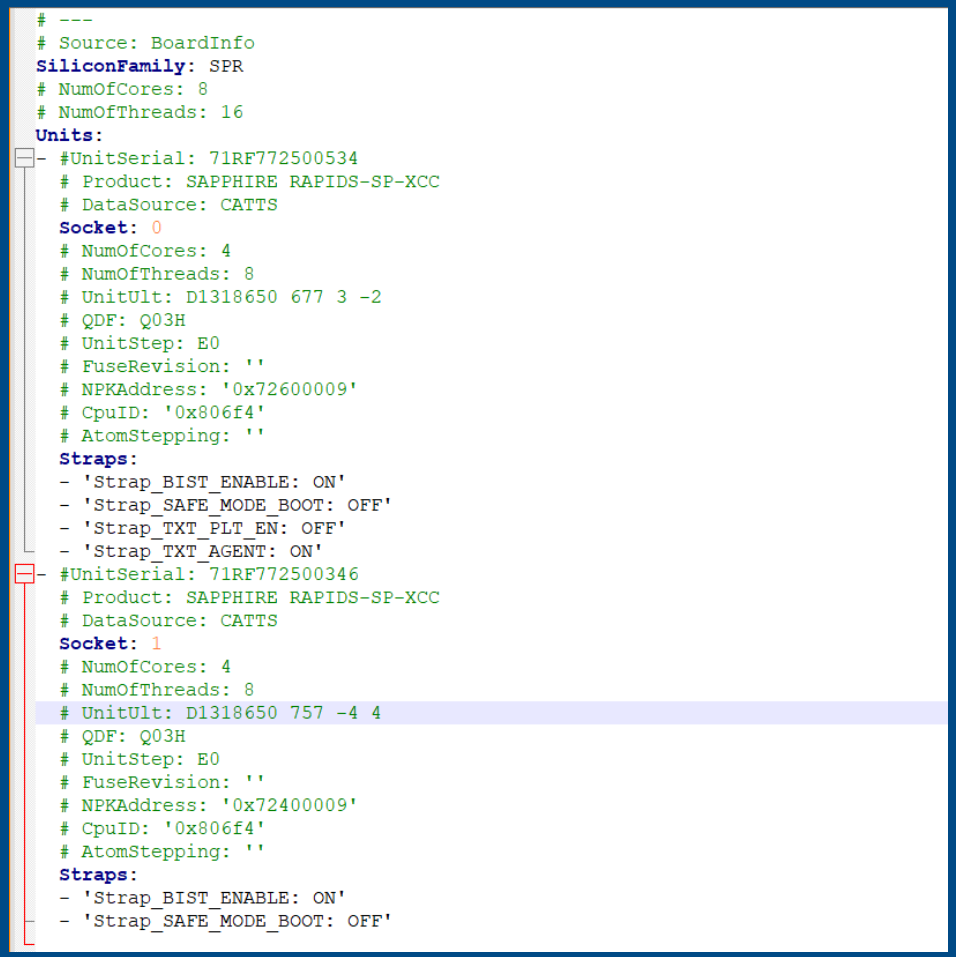
Configuration

The yaml file is mirrored from data.json. This website https://www.json2yaml.com/ can be used to convert json to yaml.

The yaml will allow an easy way to enable / disable any elements available.

Format allow for yaml file: SPR_UP.yaml, SPR_DP.yaml, EMR_UP.yaml, EMR_DP.yaml etc

```
{
    "Source": "BoardInfo",
    "SiliconFamily": "SPR",
    "NumOfCores": 8,
    "NumOfThreads": 16,
    "Units": [
        {
            "UnitSerial": "71RF772500534",
            "Product": "SAPPHIRE RAPIDS-SP-XCC",
            "DataSource": "CATTS",
            "Socket": 0,
            "NumOfCores": 4,
            "NumOfThreads": 8,
            "UnitUlt": "D1318650 677 3 -2",
            "QDF": "Q03H",
            "UnitStep": "E0",
            "FuseRevision": "",
            "NPKAddress": "0x72600009",
            "CpuID": "0x806f4",
            "AtomStepping": "",
            "Straps": [
                "Strap_BIST_ENABLE: ON",
                "Strap_SAFE_MODE_BOOT: OFF",
                "Strap_TXT_PLT_EN: OFF",
                "Strap_TXT_AGENT: ON"
            ]
        },
        {
            "UnitSerial": "71RF772500346",
            "Product": "SAPPHIRE RAPIDS-SP-XCC",
            "DataSource": "CATTS",
            "Socket": 1,
            "NumOfCores": 4,
            "NumOfThreads": 8,
            "UnitUlt": "D1318650 757 -4 4",
            "QDF": "Q03H",
            "UnitStep": "E0",
            "FuseRevision": "",
            "NPKAddress": "0x72400009",
            "CpuID": "0x806f4",
            "AtomStepping": "",
            "Straps": [
                "Strap_BIST_ENABLE: ON",
```



```
# ---
# Source: BoardInfo
SiliconFamily: SPR
# NumOfCores: 8
# NumOfThreads: 16
Units:
- #UnitSerial: 71RF772500534
  # Product: SAPPHIRE RAPIDS-SP-XCC
  # DataSource: CATTS
  Socket: 0
  # NumOfCores: 4
  # NumOfThreads: 8
  # UnitUlt: D1318650 677 3 -2
  # QDF: Q03H
  # UnitStep: E0
  # FuseRevision: ''
  # NPKAddress: '0x72600009'
  # CpuID: '0x806f4'
  # AtomStepping: ''
  Straps:
  - 'Strap_BIST_ENABLE: ON'
  - 'Strap_SAFE_MODE_BOOT: OFF'
  - 'Strap_TXT_PLT_EN: OFF'
  - 'Strap_TXT_AGENT: ON'
- #UnitSerial: 71RF772500346
  # Product: SAPPHIRE RAPIDS-SP-XCC
  # DataSource: CATTS
  Socket: 1
  # NumOfCores: 4
  # NumOfThreads: 8
  # UnitUlt: D1318650 757 -4 4
  # QDF: Q03H
  # UnitStep: E0
  # FuseRevision: ''
  # NPKAddress: '0x72400009'
  # CpuID: '0x806f4'
  # AtomStepping: ''
  Straps:
  - 'Strap_BIST_ENABLE: ON'
  - 'Strap_SAFE_MODE_BOOT: OFF'
```

Automation Conf

To enable in Xwrapper, need to add these lines in TargetConfig.xml

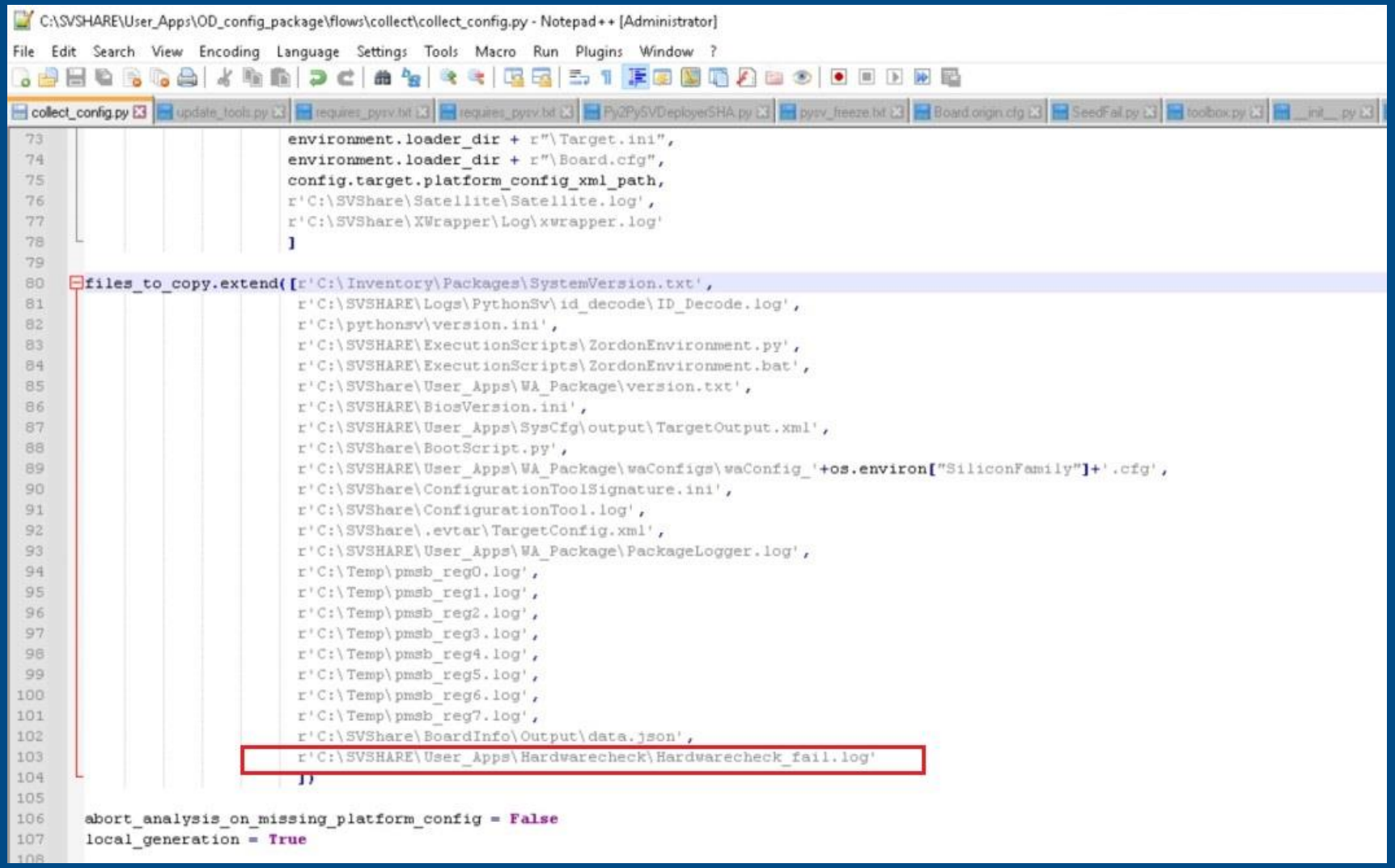
```
<CafeConfiguration>
  <CafeInterceptions>
    <Interception0>
    <Interception1>
      <Event>'PCxmlGenerated'</Event>
      <Script>'python'</Script>
      <Params>'C:\\SVShare\\BoardInfo\\BoardInfo.py --intrusive'</Params>
      <TimeoutSec>600</TimeoutSec>
    </Interception1>
    <Interception2>
    <Interception3>
    <Interception4>
    <Interception5>
    <Interception6>
    <Interception7>
    <Interception8>
      <Event>'PCxmlGenerated'</Event>
      <Script>'python'</Script>
      <Params>'C:\\SVSHARE\\User_Apps\\Hardwarecheck\\hardware_check.py --continue-on-error'</Params>
      <TimeoutSec>600</TimeoutSec>
    </Interception8>
  </CafeInterceptions>
```

To enable in Satellite, need to add these lines in board.cfg

```
[AutoDiag_POSTDone]
Enable=1
TimeOutSec = 2400
Cmd0="C:\Python27\Python.exe C:\pythonsv\skylakex\cafe\automation\F6Done.py --dis_mca_break --dis_lmup"
;Cmd1="Cwrapper.cmd skylakex\cafe\automation\cafe_hwcc.py %COMPUTERNAME%"
;Cmd2="Cwrapper.cmd skylakex\cafe\automation\cafe_swcc.py %COMPUTERNAME%"
Cmd1="C:\Python27\Python.exe C:\SVSHARE\BoardInfo\BoardInfo.py --intrusive"
Cmd2="C:\Python27\Python.exe C:\SVSHARE\User_Apps\Hardwarecheck\hardware_check.py --continue-on-error"
```

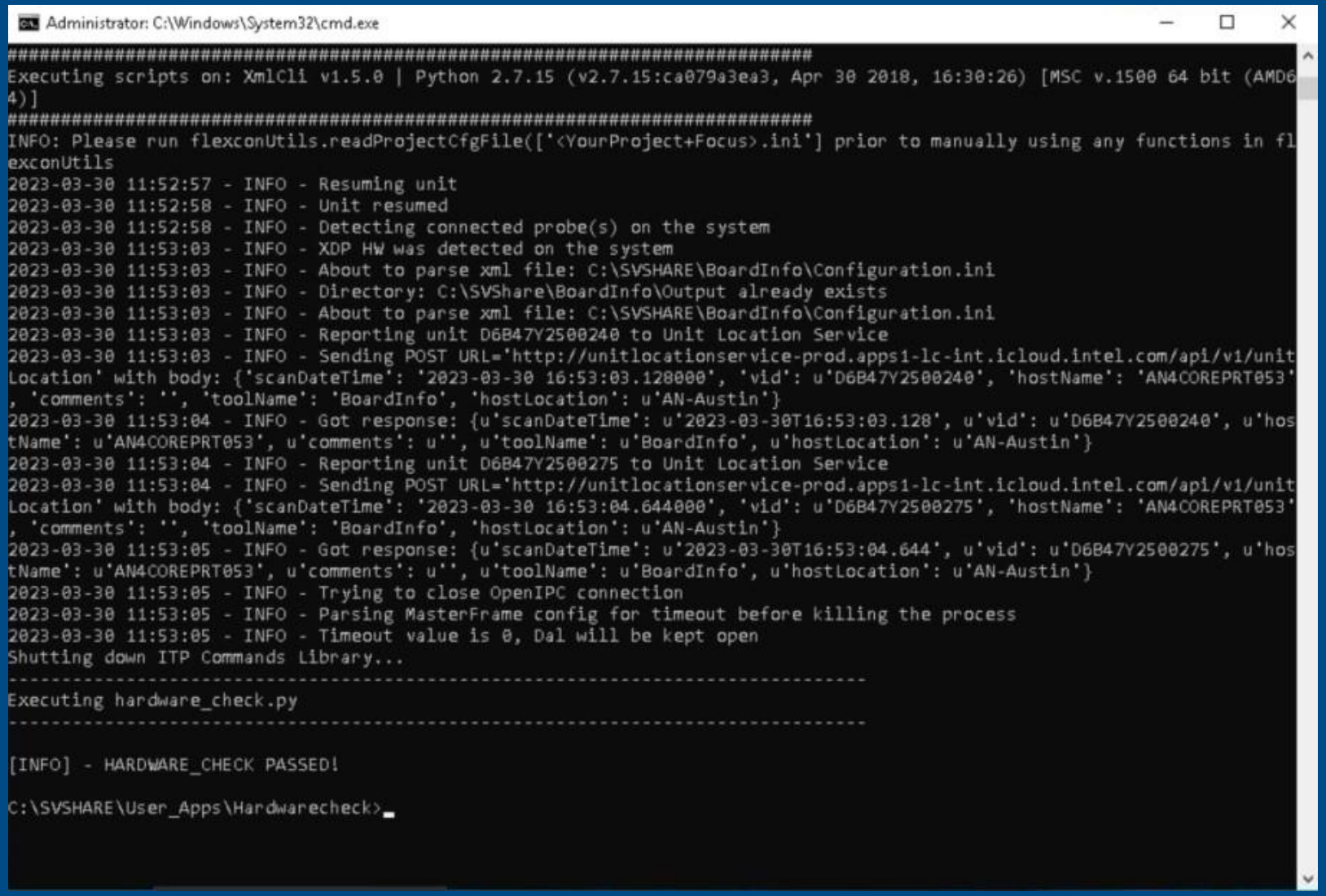
Collect Conf

To enable copy the log file to OD Collect flow during a failure. Add the path
Of the log file into collect_config.py



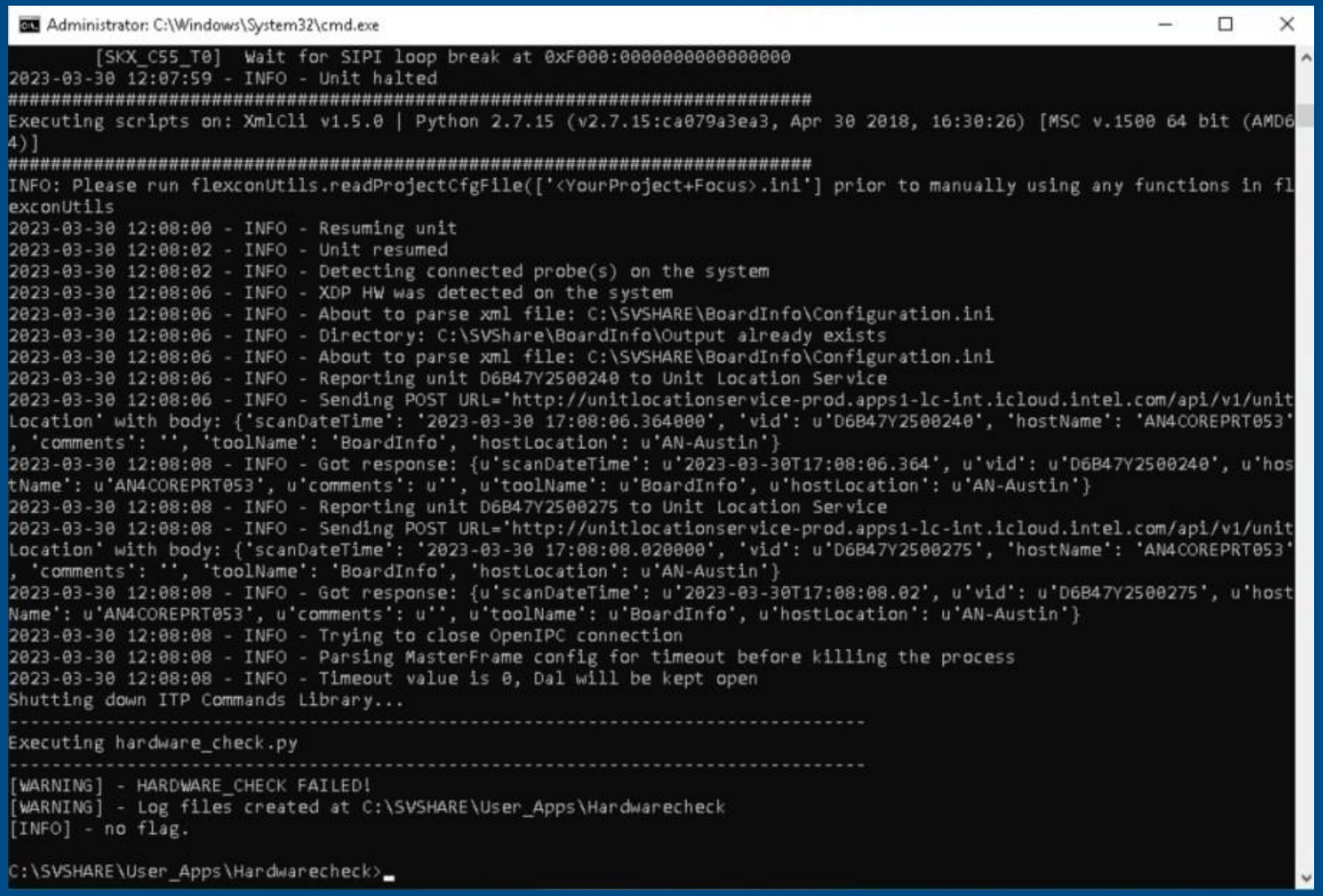
Automation

This is the output when the harwarecheck_withBoardinfo.py is run and hardware check passes:



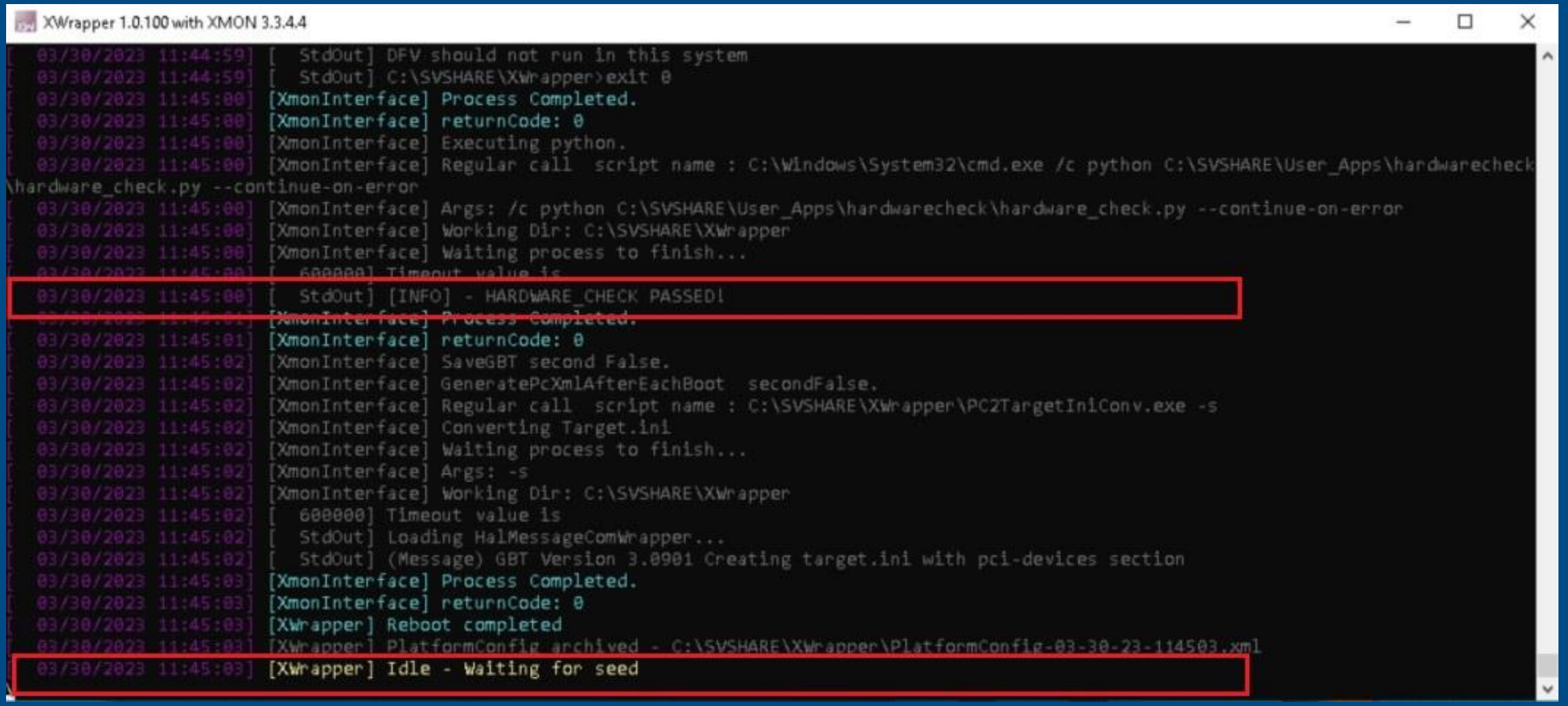
Automation

This is the output when the harwarecheck_withBoardinfo.py is run and hardware check fails:



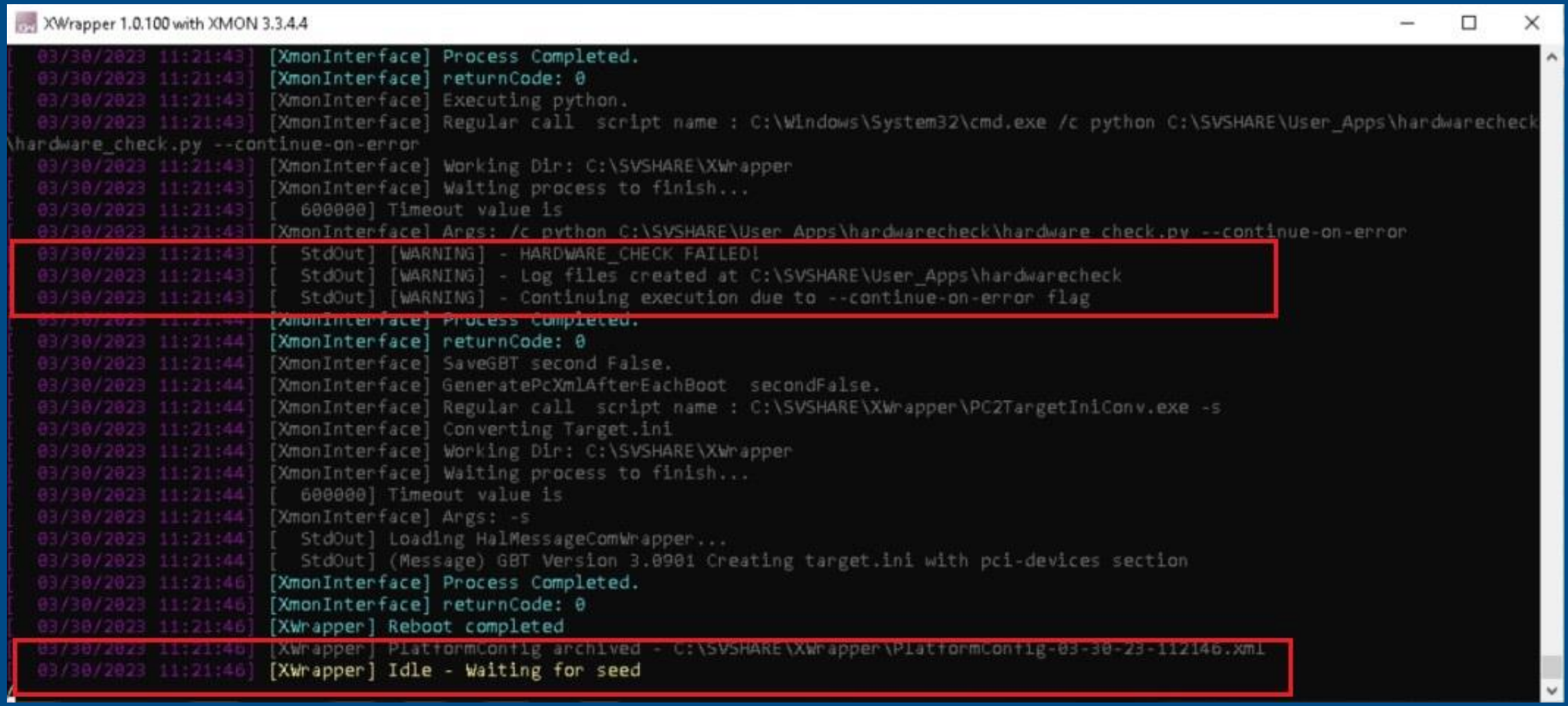
Automation

This is the output showing hardware check passing in Xwrapper:



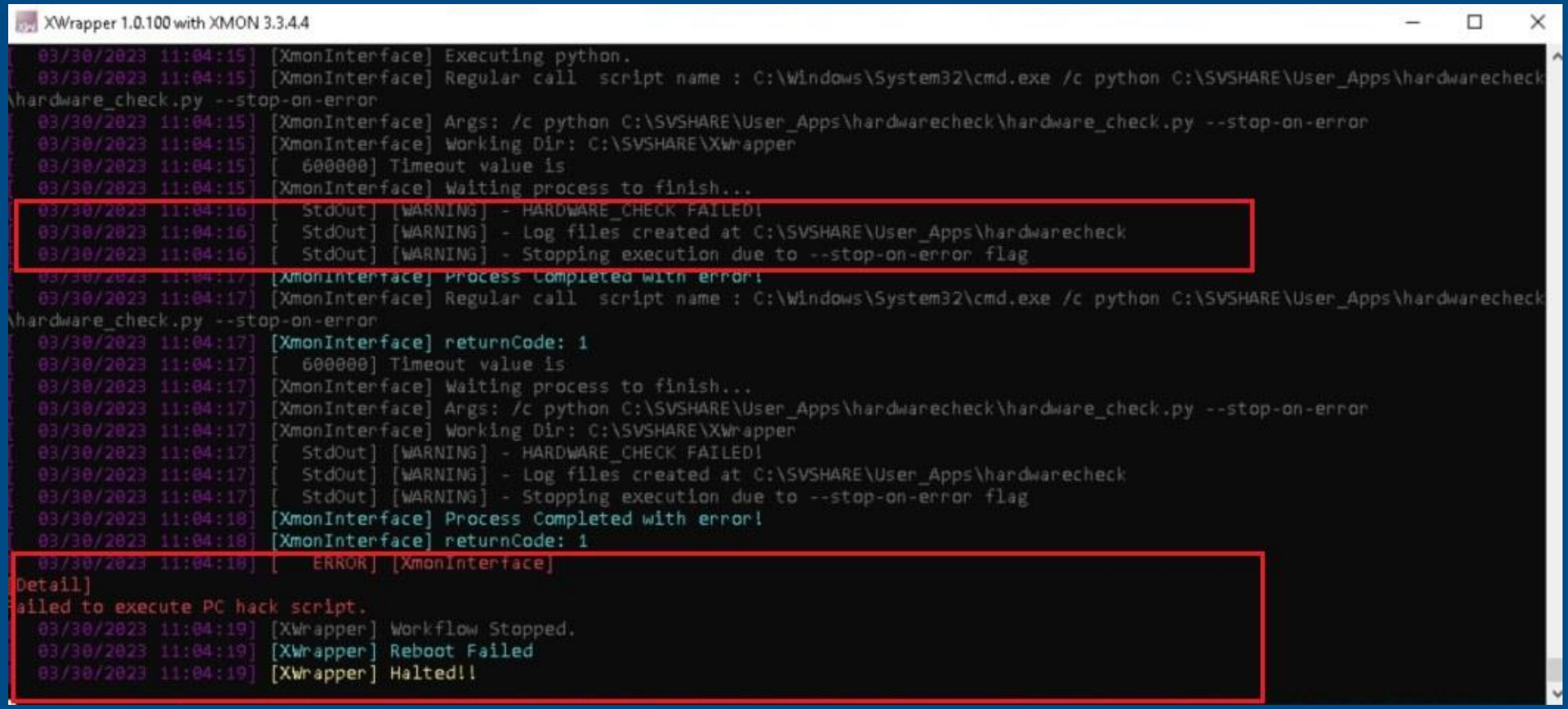
Automation

This is the output showing hardware check failed in Xwrapper with –continue-on-error flags



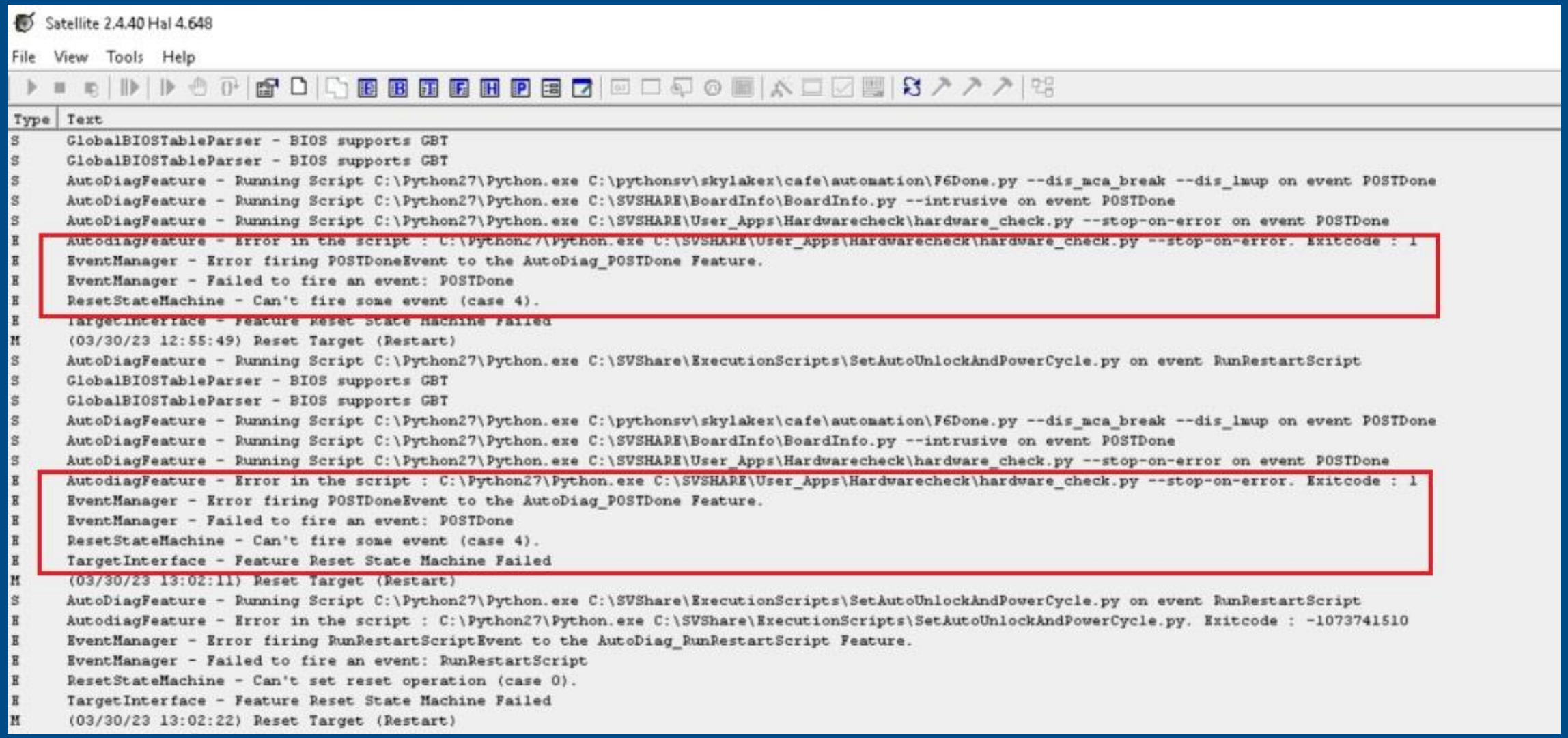
Automation

This is the output showing hardware check failed in Xwrapper with –stop-on-error flags



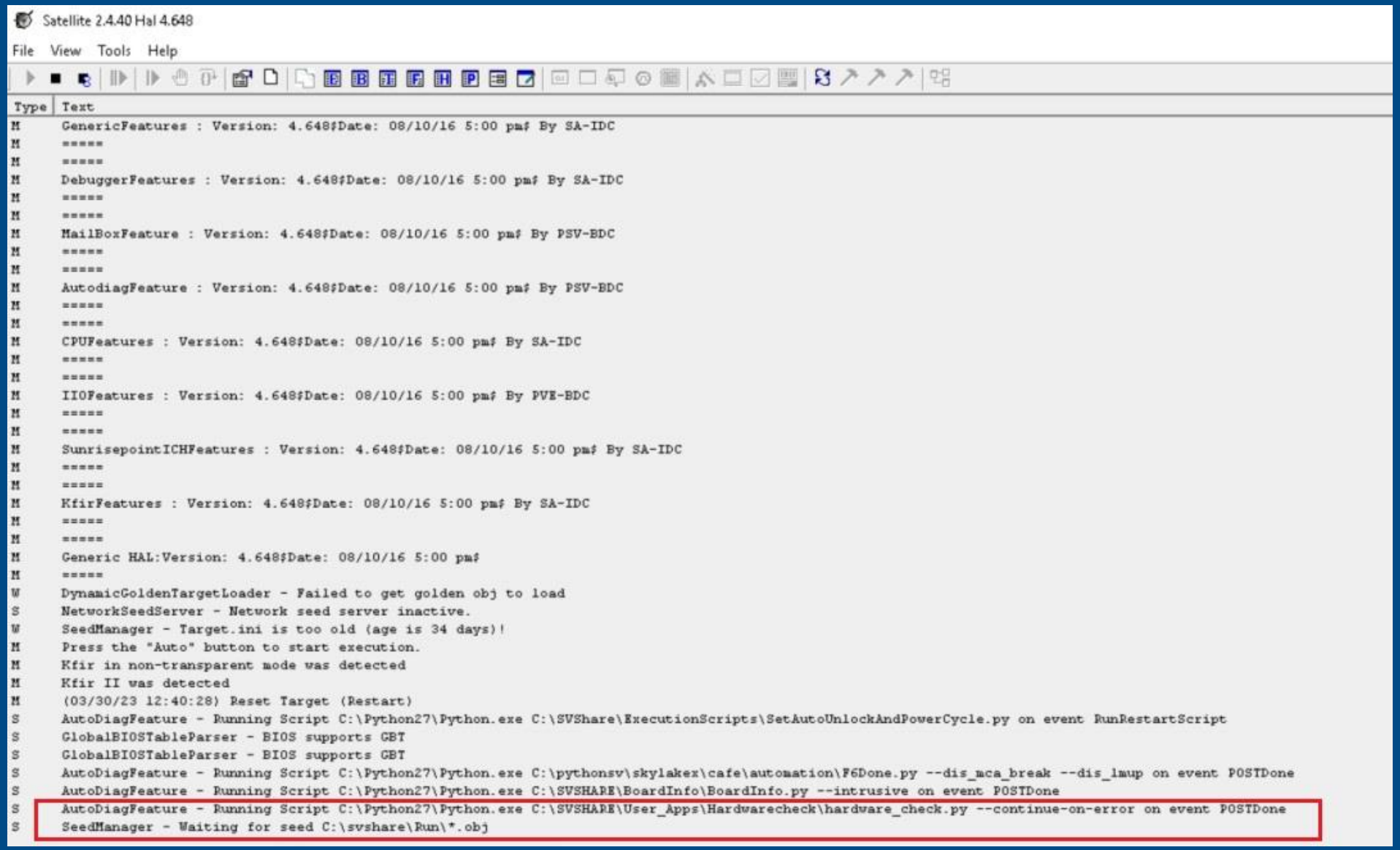
Automation

This is the output showing hardware check failed in Satellite with –stop-on-error flags



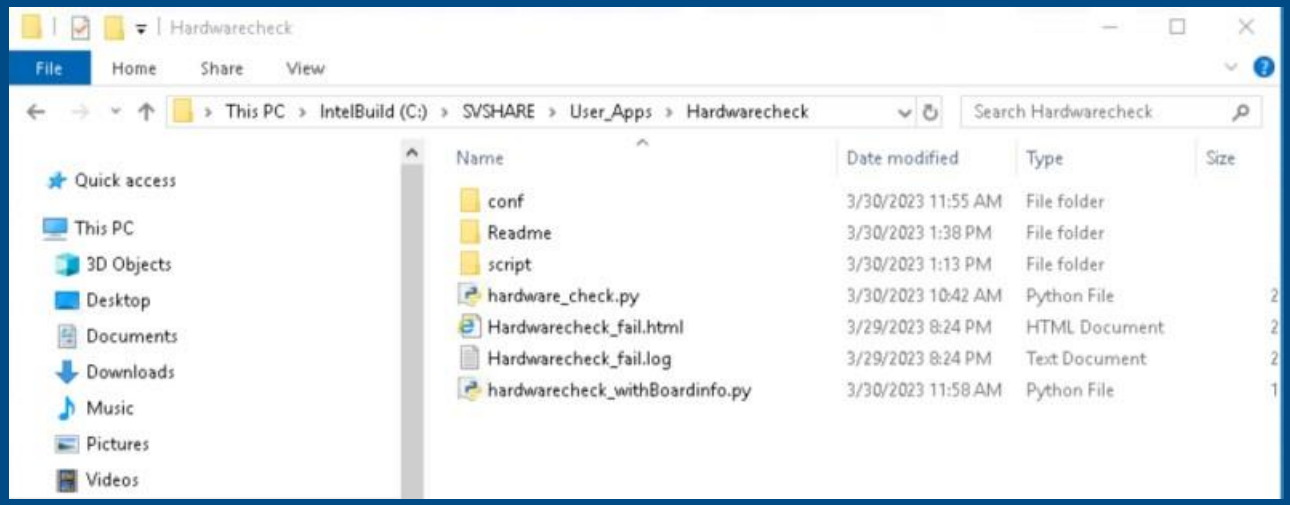
Automation

This is the output showing hardware check passed in Satellite with –continue-on-error flags



Log files

When hardware check fails, two files extensions .html and .log are created.



Hardwarcheck_fail.log

```
Differences between required SPR_DP.yaml (expected) and collected data.json (actual)
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------+
|                                Differences between required SPR_DP.yaml (expected) and collected data data.json (actual). Log created: 03/29/23 at 20:24            |
+-----------------------------------------------+-----------------------------------------------------------------+-----------------------------------------------------------------+
|                    Items                      |                          SPR_DP.yaml                            |                          data.json                              |
+-----------------------------------------------+-----------------------------------------------------------------+-----------------------------------------------------------------+
|   ['Units', 0, 'Straps', 0, 'Strap_BIST_ENABLE:    |                  Strap_BIST_ENABLE: OFF                         |                  Strap_BIST_ENABLE: ON                          |
|                    OFF']                      |                                                                 |                                                                 |
| ---------------------------------------------- | --------------------------------------------------------------- | --------------------------------------------------------------- |
+-----------------------------------------------+-----------------------------------------------------------------+-----------------------------------------------------------------+
```

Hardwarecheck_fail.html

Differences between required SPR_DP.yaml (expected) and collected data data.json (actual). Log created: 03/29/23 at 20:24

Items	SPR_DP.yaml	data.json
['Units', 0, 'Straps', 0, 'Strap_BIST_ENABLE: OFF']	Strap_BIST_ENABLE: OFF	Strap_BIST_ENABLE: ON
----------	----------	----------

Conclusion

In conclusion, the Hardware Check always calls BoardInfo to scan hardware and update the data.json file, which is then compared to the YAML configuration file.


intel