DETALYS

Hardware Testing Manual

Version: 1.5

Publication Date: 3.2.2021

Table of Contents

Introduction

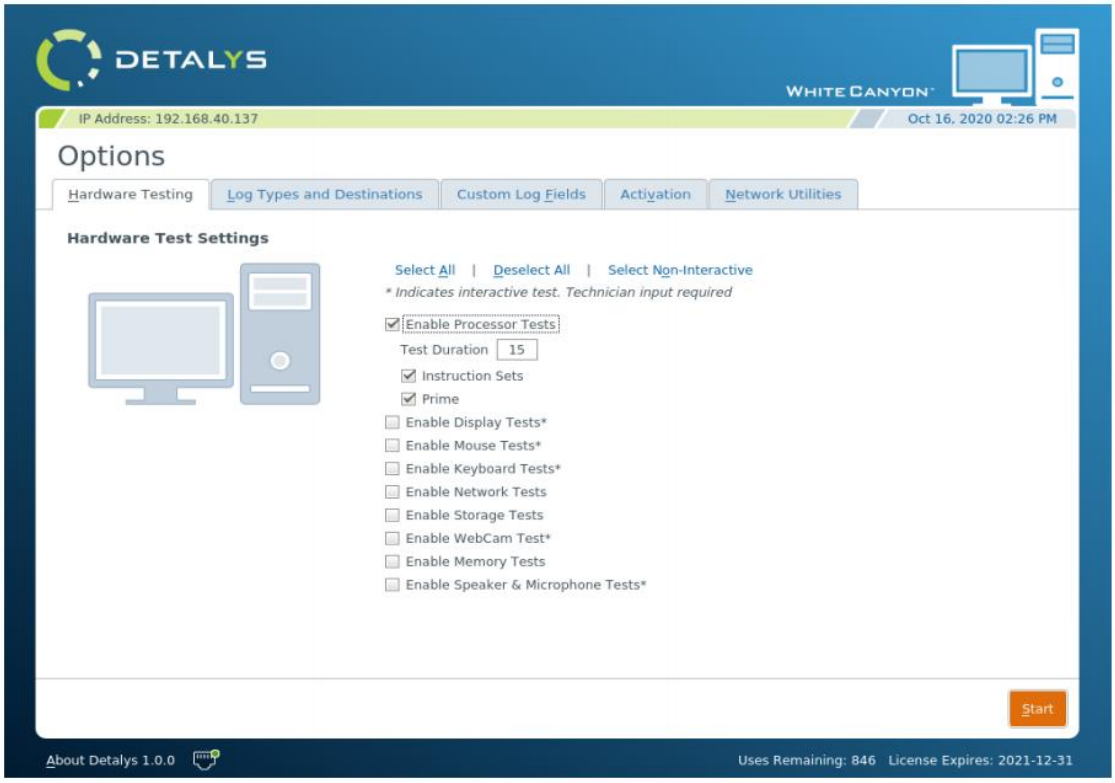
Detalys is a verbose and configurable hardware testing and diagnostics suite that provides you, the user, both automatic and interactive testing options which target all core components of the computer. These tests can play a vital role in assessing a recycled hardware asset's functionality and overall value by bringing to light any defects therein. Detalys is an essential tool that should be integrated into the asset evaluation process.

Detalys is licensed separately from WipeDrive. For reporting and diagnostic purposes, Detalys may require an active internet connection.

For questions, please reach out to us by contacting our Support department at enterprise-support@whitecanyon.com.

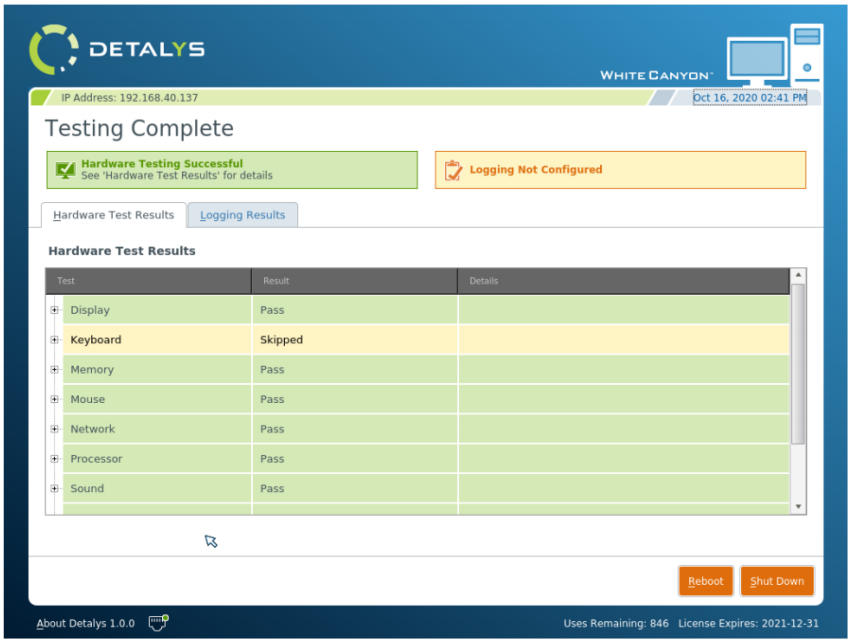
CONFIGURATION

Detalys provides many configurable options that can help the software fit into any environment. The user can enable / disable any number of hardware tests to best cover all aspects of the device being tested. Audit log reporting can be configured with the desired log file types and logging destinations, providing the user flexibility

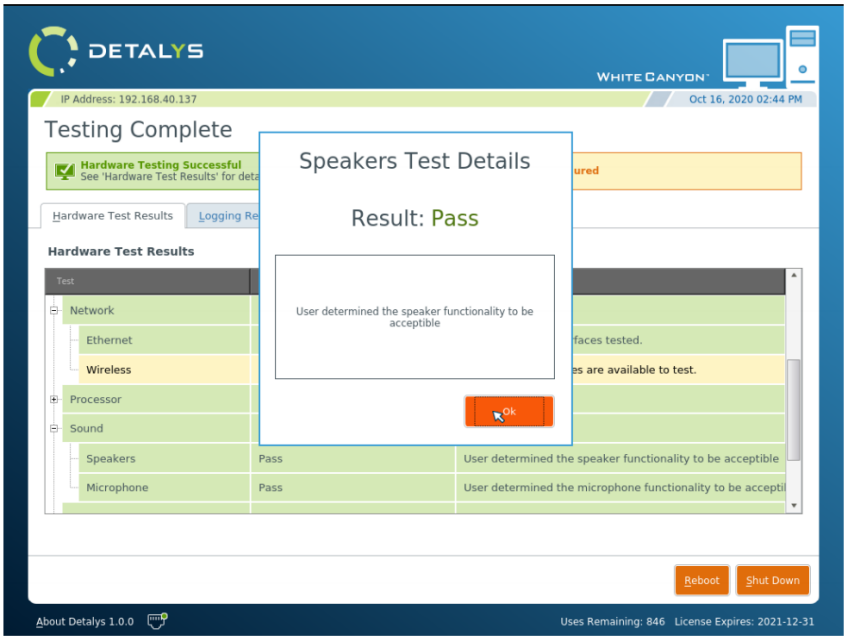


TESTING RESULTS

Once all hardware tests have been run to completion, the results are displayed to the user. Each hardware test category is listed along side an overall result for that category. Logging results will be listed here on this page as well.



If additional information regarding a specific hardware test is wanted, clicking on the '+' icon found on the left side of a test category will expand the list item to show all the subtests for the category chosen. Clicking on any subtest item will show a pop-up with all the details for that subtest as well.



Hardware Item Tests

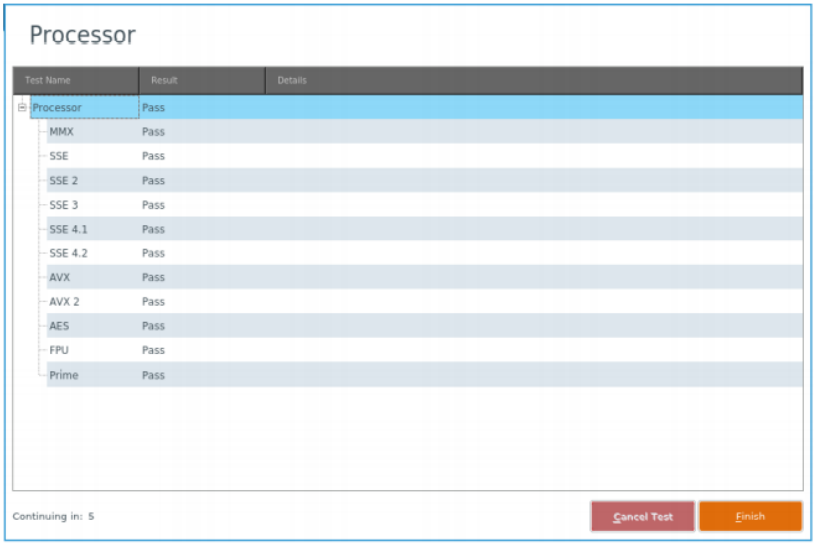
PROCESSOR TEST

The processor tests are designed to vigorously test your machine's CPU by testing a large subset of operations for each supported family of processor commands, being inclusive of all registers and areas of the processing unit. By default, the configuration will test all supported command set families and registers for a given duration but alternatively can be configured to run tests for specific families (when supported). **Please note:** these tests do not run on 32-bit processors and are only accessible on 64-bit builds.

CPU Sub Test Categories:

- Matrix Math Extension (MMX)

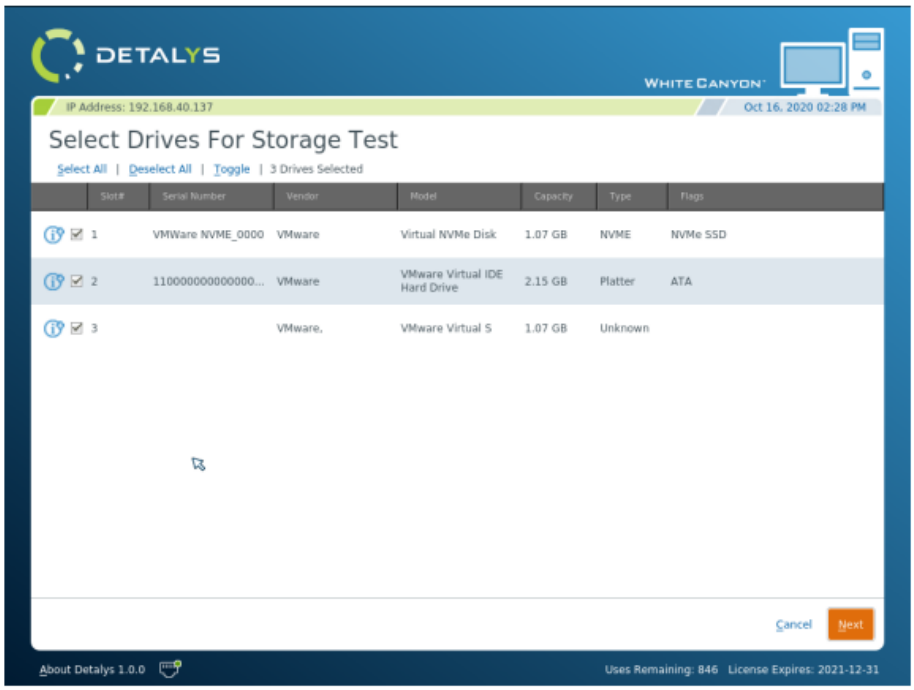
- Streaming SIMD Extension (SSE)
- SSE v. 2
- SSE v. 3
- SSE v. 4.1
- SSE v. 4.2
- Advanced Vector Extension (AVX)
- AVX v. 2
- Floating Point Unit (FPU)
- Prime Number Generator
- Advanced Encryption Standard (AES)

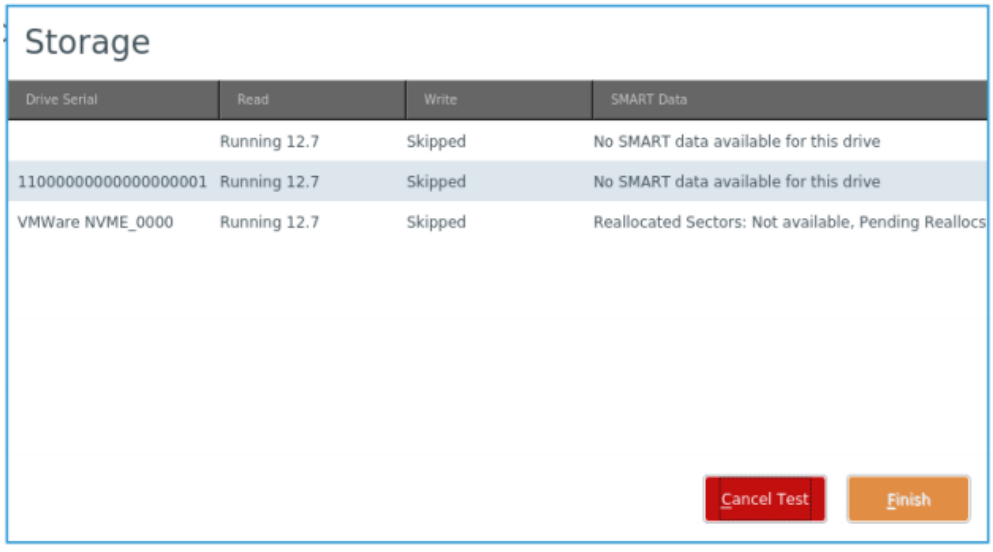


The processor tests are non-interactive and will engage the CPU for the length of a specified duration. Each enabled sub-test category will be run in succession of each other, and results will be displayed upon all the tests finishing. If a test fails, details can be viewed by clicking on the corresponding row of the failed test. An automatic countdown will start and will automatically move on with the other selected hardware tests, unless a test item is clicked for additional details.

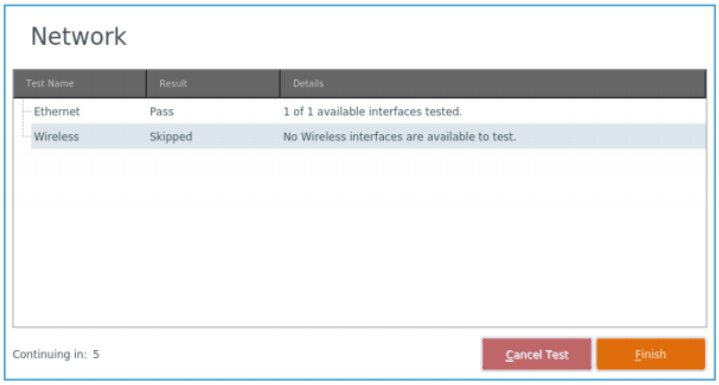
STORAGE TEST

The storage tests are designed to run timed operations to test a hard drive's read and write capabilities and will also query the hard drive for its overall SMART health status. This non-interactive test will, by default, perform non-destructive test operations on the drives selected, thus preserving the data that resides on the device. A data destructive version of the test can be enabled if this functionality is desired. SMART health status will only be recorded when the queried hard drive supports it.





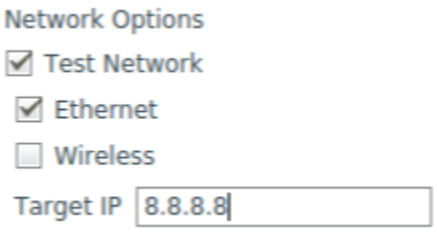
NETWORK TEST



The network tests are designed to detect and test all active interfaces found of a specific type. This set of tests requires either a physical Ethernet connection, or that the Wi-Fi parameters have already been entered and configured prior to the test running. The Network tests are also non-interactive and will run automatically when configured. Once all interfaces have been detected, each interface will be tested for connectivity to the nearest Gateway on the network. An interface will show a failure if this operation cannot be performed. If a test fails, details can be viewed by clicking on the corresponding row of the failed test. An automatic countdown will start and will automatically move on with the other selected hardware tests, unless a test item is clicked for additional details.

The Wireless network test can be configured to scan for available SSID's instead of performing a connectivity test, removing the need for prior Wi-Fi configuration. An additional search option can be supplied to further configure this testing mode so that it will search for specific SSID's that match the given search parameters. This search option supports the entry of any valid regular expression.

Some networking devices are configured to ignore any ICMP echo request/reply commands, which will cause an interface test to fail. If a desired IP destination address is known, it can be entered via the settings page so that this address will be used for connectivity testing, rather than the default nearest gateway.

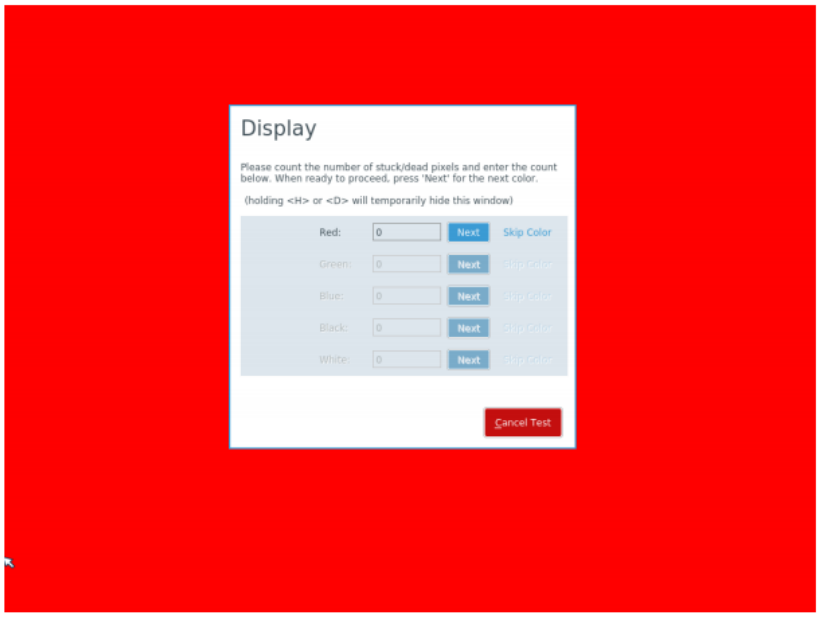


Please note: Wi-Fi is only accessible if the machine running the program has a Wireless Internet card that has drivers which are supported in Linux.

DISPLAY TEST

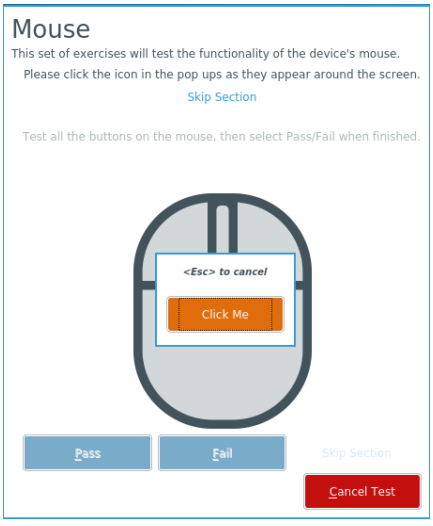
The display tests will help determine if there are any stuck/dead pixels in the device's screen or monitor. The entire screen will display five colors, one by one, requiring a stuck/dead pixel count to be entered for each color. Those colors are red, green, blue, black, and white.

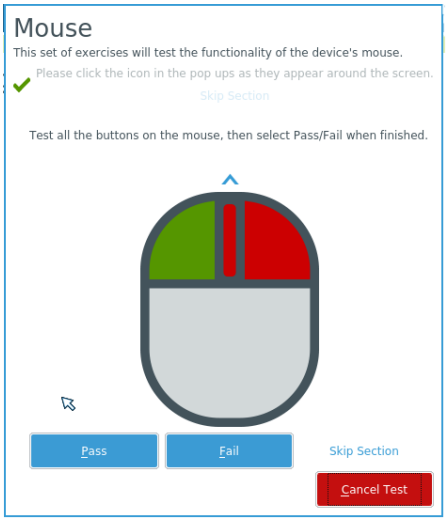
Keep in mind that the pop-up window can be moved by clicking and dragging it, allowing the entire screen to be thoroughly checked. Holding the keys <H> or <D> on the keyboard can also be used to hide the pop-up window when a mouse isn't present.



MOUSE TEST

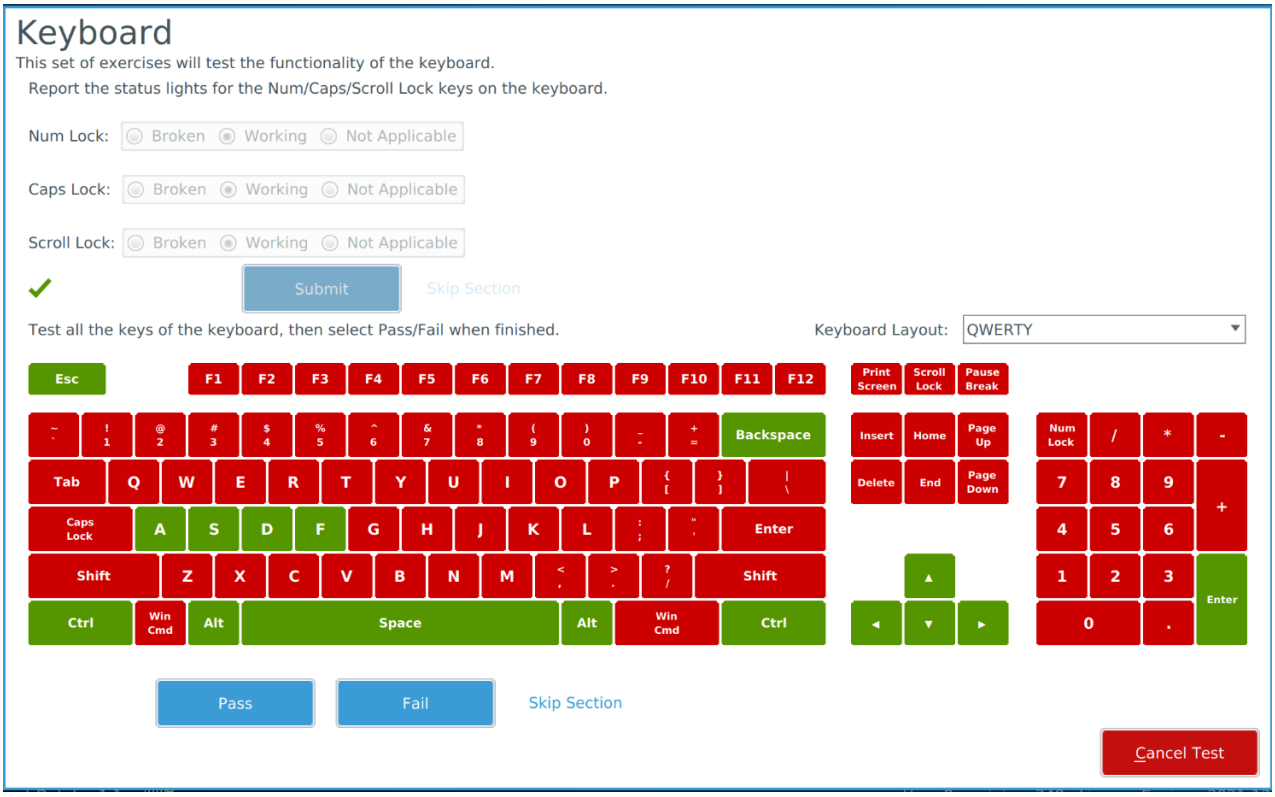
The mouse tests will check all major functions of the device's mouse. This interactive test has two parts. The first part will test the mouse's ability to reach all parts of the screen, making sure that the movement is acceptable. If any section of the screen is unreachable, pressing the 'Esc' key will cancel the test and mark the test as failed. The second part will test the five major buttons/functions on the mouse, namely 'Left Click', 'Right Click', 'Scroll Wheel Click', 'Scroll Up', and 'Scroll Down'. If any mouse buttons aren't working as expected, then the test can be marked as failed by clicking 'Fail'. If everything is working properly, then 'Pass' can be selected.





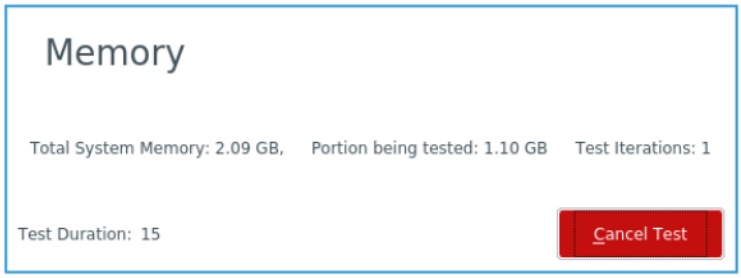
KEYBOARD TEST

The keyboard tests are broken up into two interactive parts that test all major parts of a device's keyboard. The lights test will test the status indicator lights for the 'Caps Lock', 'Num Lock', and 'Scroll Lock' keys. The general test is designed to be a free-form test, where you can test any subset of keys on the keyboard and can then give an overall evaluation of the device's functionality based on the results seen during the test. Any combination of these tests can be configured to run when the keyboard tests are enabled.



MEMORY TEST

The memory test is designed to test the storage capability and accuracy of the entirety of all available random-access memory (RAM) of the system. It is important to note that the Detalys program runs entirely within the system's RAM so the amount of memory being tested by Detalys will not equate to the full memory stick capacity of the device. The test will run for a specified duration and will complete as many repetitions on the memory as possible within the given duration. This helps stress the device to ensure that the test is thorough in detecting any errors that may reside. If a repetition takes longer to complete than the set duration for the test, the repetition will be run to completion before the test is exited.

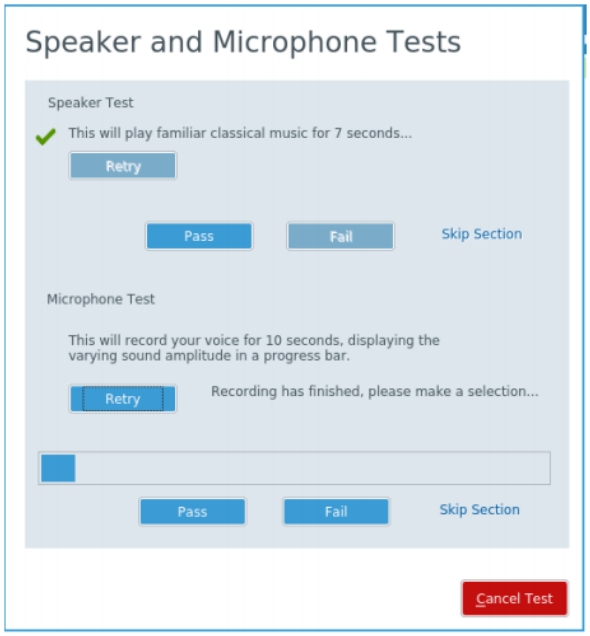


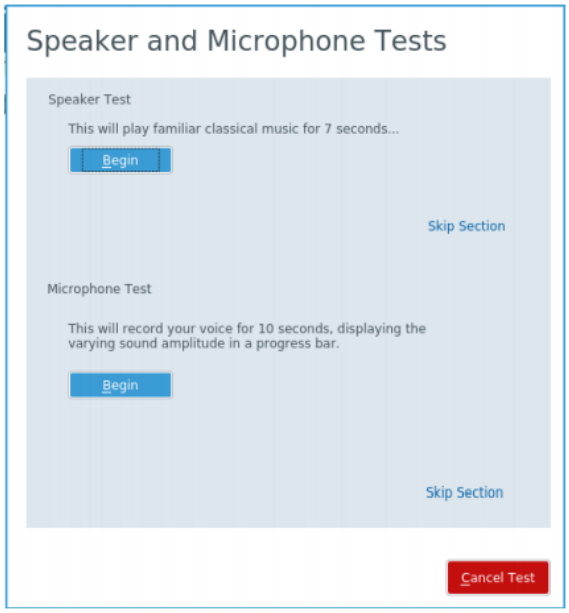
WEBCAM TEST

Most portable computer and laptop devices come with a web camera built into the screen. The webcam test helps the user get a snapshot of the cameras current state and gives the user a quick glimpse at the webcam's picture quality. Once the snapshot is taken, the captured photo will be displayed on screen for evaluation. The snapshot can be retaken as many times as desired and will ultimately be passed or failed by the user.

SPEAKER AND MICROPHONE TEST

The speaker and microphone tests allow the user to test the audio playback and capture capabilities of the device being tested. The speaker test will play a music recording and prompt the user for a pass/fail decision based on the audio quality.



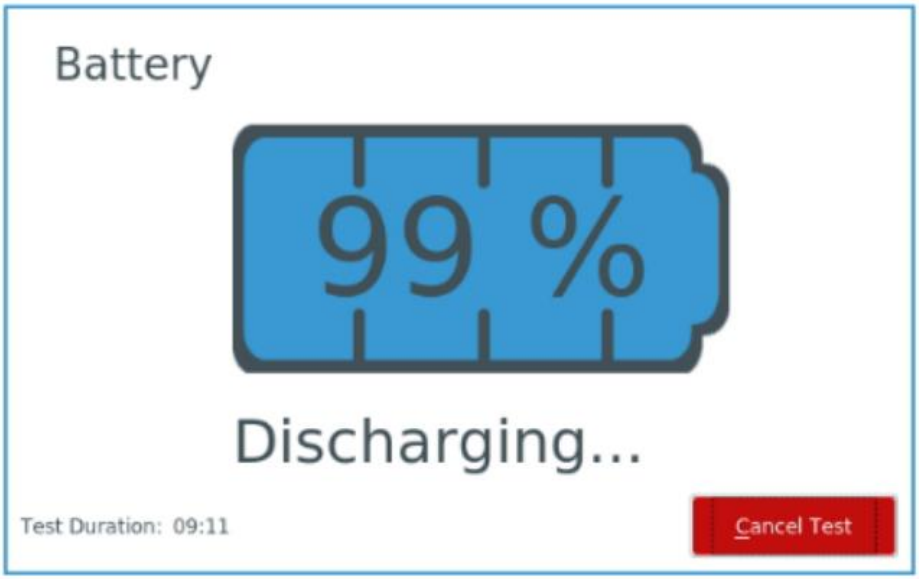


The microphone test will listen for the user's voice and display in a bar the level at which it could pick up the voice. The user is then prompted to make a pass/fail decision based on the performance seen with the microphones ability to pick up sound.

BATTERY TEST

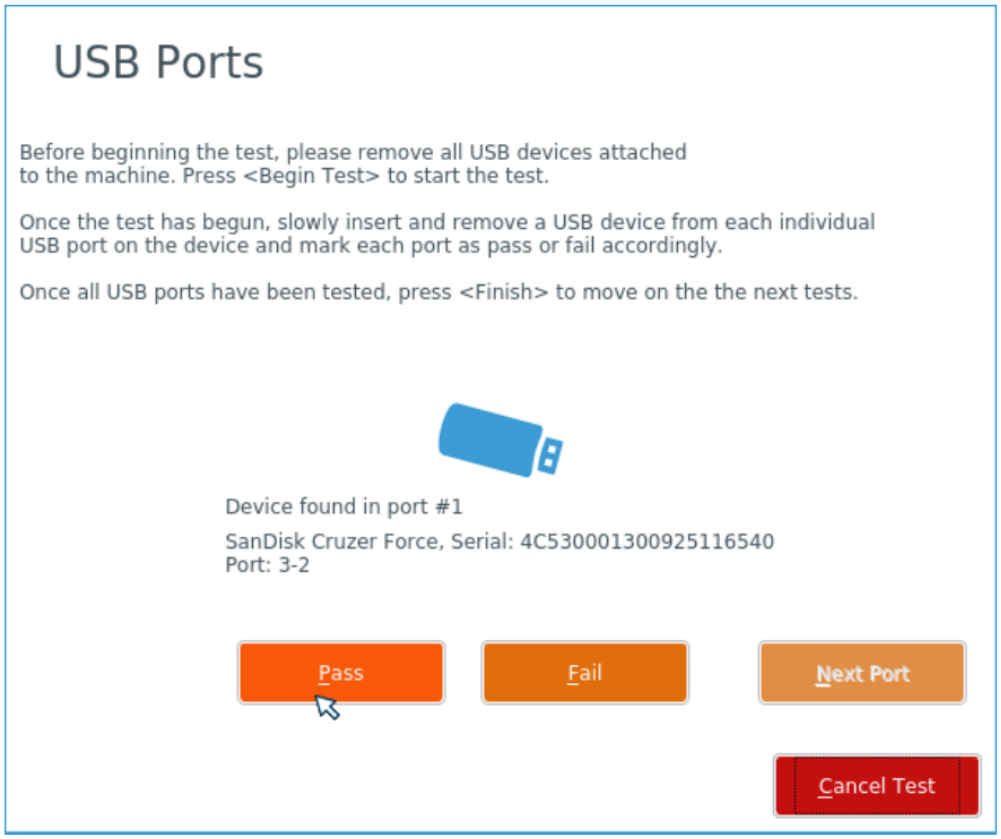
The battery test helps the user diagnose weak batteries in the devices being tested. The test will put stressful loads on the system's memory and CPU components for a set duration in attempt to cause higher power draw on the battery. The total charge expenditure is tracked over the length of the test and is then evaluated against the set threshold of percent charge lost that is allowed for the test.

The battery test requires a minimum charge of 50% before the test can begin. If the system's battery doesn't have sufficient charge, the battery will need to be charged for a small duration before the test begins. This duration, as well as the discharge test duration can be modified via the settings screen.



USB PORTS TEST

The USB Ports test will help the user determine the functionality of all USB-1, USB-2, USB-3, and USB-C type ports found on the device being tested. Before the test is started, all USB devices should be removed from the system. Once the test has begun, the user will attach a USB storage device to a port on the device. When the device is recognized, the user interface will indicate that a device was attached. The device can be removed from the port and the interface should indicate the removal of the device as well. After these actions have been performed for the port being tested, the port can be graded as either pass or fail. If there are more ports to test, the user will select the 'Next Port' button and then will repeat the same process previously mentioned on the new port. Once this has been done for every USB port available on the device, pressing 'Finish' will conclude the test and submit the results.



Audit Logs

For general information regarding the audit log contents and logging methods that are supported, please refer to the WipeDrive Logging Manual. Below are examples showcasing the additional information that Detalys provides in the supported audit log formats.

XML

```
<HardwareTest>
    <Processor>
      <MMX>
        <Result>Pass</Result>
        <Notes></Notes>
      </MMX>
      <SSE>
        <Result>Pass</Result>
        <Notes></Notes>
      </SSE>
      <SSE2>
        <Result>Pass</Result>
        <Notes></Notes>
      </SSE2>
      <SSE3>
        <Result>Pass</Result>
        <Notes></Notes>
      </SSE3>
      <SSE4.1>
        <Result>Pass</Result>
        <Notes></Notes>
      </SSE4.1>
      <SSE4.2>
        <Result>Pass</Result>
        <Notes></Notes>
      </SSE4.2>
      <AVX>
        <Result>Pass</Result>
        <Notes></Notes>
      </AVX>
      <AVX2>
        <Result>Pass</Result>
        <Notes></Notes>
      </AVX2>
      <Prime>
        <Result>Pass</Result>
        <Notes></Notes>
      </Prime>
      <FPU>
        <Result>Pass</Result>
        <Notes></Notes>
      </FPU>
      <AES>
        <Result>Pass</Result>
        <Notes></Notes>
      </AES>
    </Processor>
    <Display>
      <Red>
        <Result>Pass</Result>
        <Notes>0 px</Notes>
      </Red>
      <Green>
        <Result>Pass</Result>
        <Notes>0 px</Notes>
      </Green>
      <Blue>
        <Result>Pass</Result>
        <Notes>0 px</Notes>
      </Blue>
      <Black>
        <Result>Pass</Result>
        <Notes>0 px</Notes>
      </Black>
      <White>
        <Result>Pass</Result>
        <Notes>0 px</Notes>
      </White>
      <TotalDeadPixels>
        <Result>Pass</Result>
        <Notes>0 px</Notes>
      </TotalDeadPixels>
    </Display>
    <Mouse>
```

```
    <Movement>
      <Result>Skipped</Result>
      <Notes></Notes>
    </Movement>
    <Buttons>
      <Result>Pass</Result>
      <Notes></Notes>
    </Buttons>
  </Mouse>
  <Keyboard>
    <General>
      <Result>Pass</Result>
      <Notes>User determined keyboard functionality to be acceptable</Notes>
    </General>
    <Lights>
      <Result>Pass</Result>
      <Notes>All status lights are working correctly</Notes>
    </Lights>
  </Keyboard>
  <Network>
    <Ethernet>
      <Result>Pass</Result>
      <Notes>1 of 1 available interfaces tested</Notes>
    </Ethernet>
    <Wireless>
      <Result>Fail</Result>
      <Notes>0 of 1 available interfaces tested</Notes>
    </Wireless>
  </Network>
  <Storage>
    <Read>
      <Result>Pass</Result>
      <Notes></Notes>
    </Read>
    <Write>
      <Result>Pass</Result>
      <Notes></Notes>
    </Write>
    <SMARTHealth>
      <Result>Skipped</Result>
      <Notes>S35ENX0J616416 - SMART Health Skipped</Notes>
    </SMARTHealth>
  </Storage>
  <Memory>
    <Memory>
      <Result>Pass</Result>
      <Notes>6.2GB, 5 repetitions, runtime: 6 seconds</Notes>
    </Memory>
  </Memory>
  <WebCam>
    <WebCamera>
      <Result>Pass</Result>
      <Notes></Notes>
    </WebCamera>
  </WebCam>
  <Sound>
    <Speakers>
      <Result>Pass</Result>
      <Notes></Notes>
    </Speakers>
    <Microphone>
      <Result>Pass</Result>
      <Notes></Notes>
    </Microphone>
  </Sound>
</HardwareTest>
```

JSON


```
"HardwareTests": {
    "TimeStarted": 0,
    "Display": [
      {
        "Name": "Red",
        "Notes": "0 px",
        "Result": 0
      },
      {
        "Name": "Green",
        "Notes": "0 px",
        "Result": 0
      },
      {
        "Name": "Blue",
        "Notes": "0 px",
        "Result": 0
      },
      {
        "Name": "Black",
        "Notes": "0 px",
        "Result": 0
      },
      {
        "Name": "White",
        "Notes": "0 px",
        "Result": 0
      },
      {
        "Name": "Total Dead Pixels",
        "Notes": "0 px",
        "Result": 0
      }
    ],
    "Mouse": [
      {
        "Name": "Movement",
        "Notes": "",
        "Result": 2
      },
      {
        "Name": "Buttons",
        "Notes": "",
        "Result": 0
      }
    ],
    "Keyboard": [
      {
        "Name": "General",
        "Notes": "User determined keyboard functionality to be acceptable",
        "Result": 0
      },
      {
        "Name": "Lights",
        "Notes": "All status lights are working correctly",
        "Result": 0
      }
    ],
    "Processor": [
      {
        "Name": "MMX",
        "Notes": "",
        "Result": 0
      },
```

```
    {
      "Name": "SSE",
      "Notes": "",
      "Result": 0
    },
    {
      "Name": "SSE 2",
      "Notes": "",
      "Result": 0
    },
    {
      "Name": "SSE 3",
      "Notes": "",
      "Result": 0
    },
    {
      "Name": "SSE 4.1",
      "Notes": "",
      "Result": 0
    },
    {
      "Name": "SSE 4.2",
      "Notes": "",
      "Result": 0
    },
    {
      "Name": "AVX",
      "Notes": "",
      "Result": 0
    },
    {
      "Name": "AVX 2",
      "Notes": "",
      "Result": 0
    },
    {
      "Name": "Prime",
      "Notes": "",
      "Result": 0
    },
    {
      "Name": "FPU",
      "Notes": "",
      "Result": 0
    },
    {
      "Name": "AES",
      "Notes": "",
      "Result": 0
    }
  ],
  "Network": [
    {
      "Name": "Ethernet",
      "Notes": "1 of 1 available interfaces tested",
      "Result": 0
    },
    {
      "Name": "Wireless",
      "Notes": "0 of 1 available interfaces tested",
      "Result": 1
    }
  ],
  "Storage": [
    {
      "Name": "Read",
      "Notes": "",
      "Result": 0
```

```
      },
      {
        "Name": "Write",
        "Notes": "",
        "Result": 0
      },
      {
        "Name": "SMART Health",
        "Notes": "S35ENX0J616416 - SMART Health Skipped",
        "Result": 2
      }
    ],
    "Ram": [
      {
        "Name": "Memory",
        "Notes": "6.2GB, 5 repetitions, runtime: 6 seconds",
        "Result": 0
      }
    ],
    "Webcam": [
      {
        "Name": "Web Camera",
        "Notes": "",
        "Result": 0
      }
    ],
    "Sound": [
      {
        "Name": "Speakers",
        "Notes": "",
        "Result": 0
      },
      {
        "Name": "Microphone",
        "Notes": "",
        "Result": 0
      }
    ]
}
```

PDF

DETALYS

Operation Info

Action Result	SUCCESS
Software Used	Detalys 1.0.0
Kernel Version	5.6.13-gentoo-x86_64
Job UUID	7628CDC3-1CE2-4138-B234-8BEF005E0719
Action Start Time	Monday, 19 Oct 2020 at 18:05:29
Action Duration	00:02:24

Hardware Information

Computer	Vendor: LENOVO Model: 20HGS17V00 Serial: PC0MKUYL
Motherboard	Vendor: LENOVO Model: 20HGS17V00
Processor	Intel(R) Core(TM) i5-7300U CPU @ 2.60GHz
RAM	8192 MB, Memory banks filled 2/2, DDR4 SODIMM
TPM	Found TPM 2.0
NIC	Intel Corporation, Wireless 8265 / 8275 MAC Address: f8:59:71:49:35:e0
NIC	Intel Corporation, Ethernet Connection (4) I219-LM MAC Address: 54:e1:ad:04:49:39
Drive 1	Vendor: Samsung Electronics Co Ltd Model: SAMSUNG MZVLW256HEHP-000L7 Serial: S35ENX0J616416 Size: 256.06 GB Drive Media Type: NVME Firmware Version: 4L7QCXB7 EUI64: 002538b671b0b465 Pre-wipe SMART Health Status: PASSED

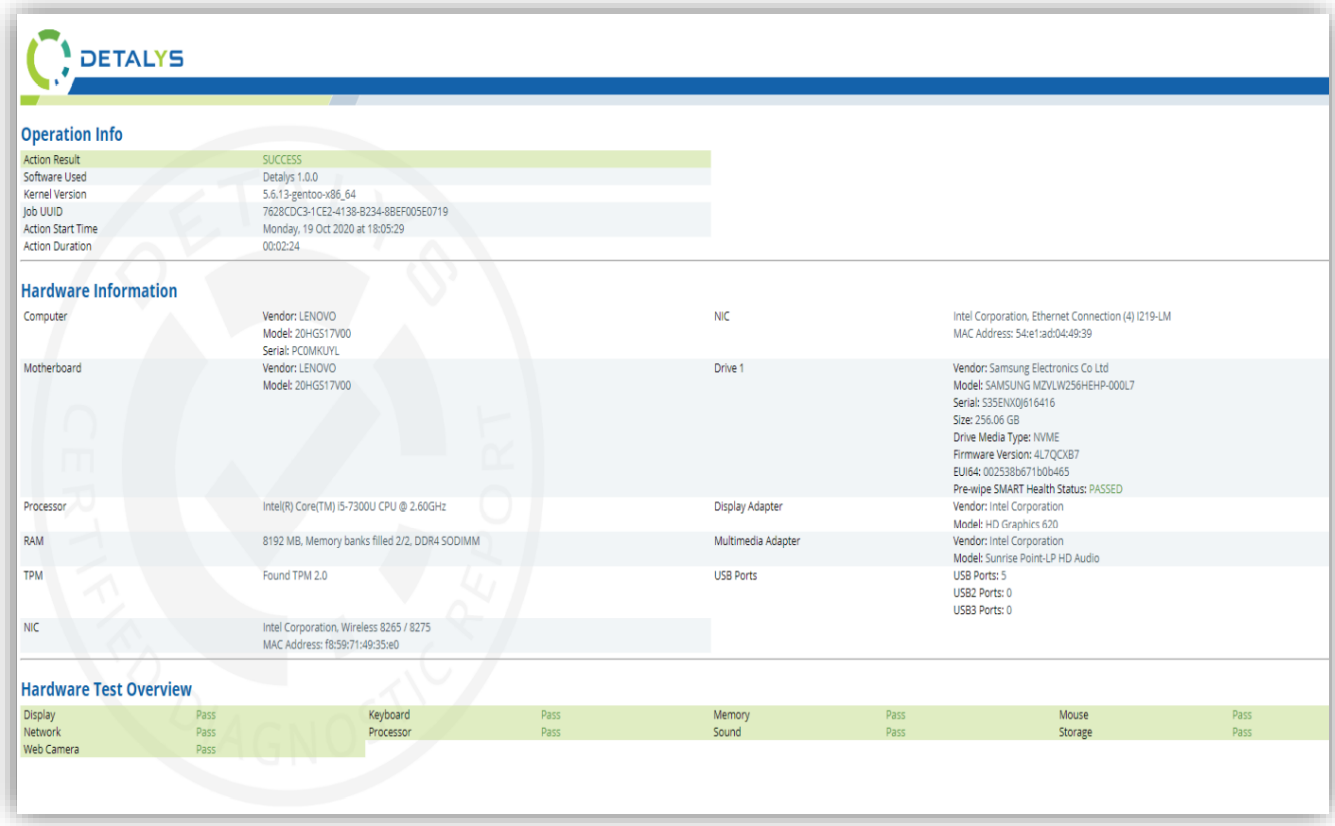
Display Adapter	Vendor: Intel Corporation Model: HD Graphics 620
Multimedia Adapter	Vendor: Intel Corporation Model: Sunrise Point-LP HD Audio
USB Ports	USB Ports: 5 USB2 Ports: 0 USB3 Ports: 0

Hardware Test Overview

Display	Pass	Keyboard	Pass	Memory	Pass	Mouse	Pass
Network	Pass	Processor	Pass	Sound	Pass	Storage	Pass
Web Camera	Pass						

Page 1/1

HTML



DETALYS

Operation Info

Action Result	SUCCESS
Software Used	Detalys 1.0.0
Kernel Version	5.6.13-gentoo-x86_64
Job UUID	7628CDC3-1CE2-4138-B234-8BEF005E0719
Action Start Time	Monday, 19 Oct 2020 at 18:05:29
Action Duration	00:02:24

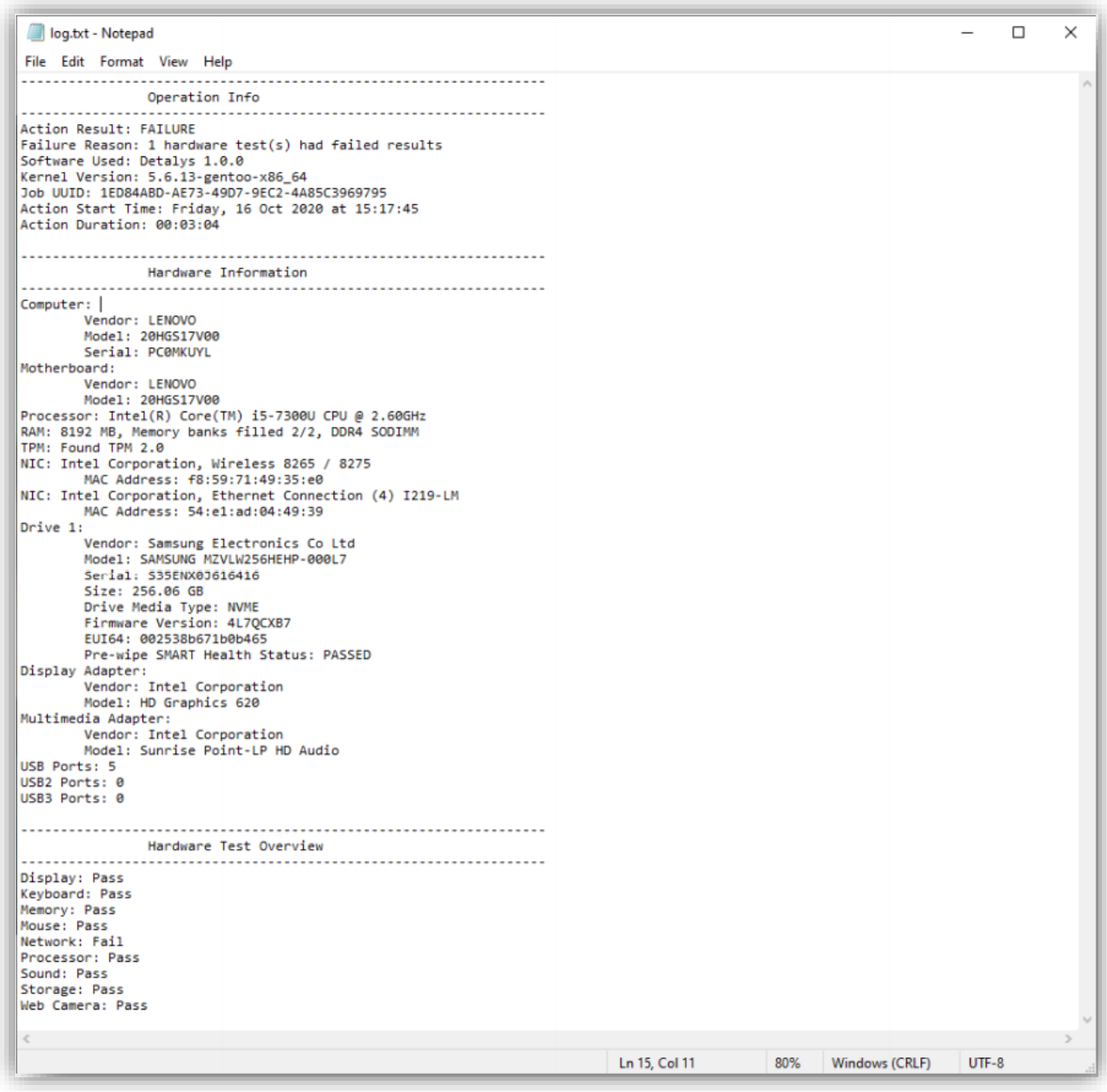
Hardware Information

Computer	Vendor: LENOVO Model: 20HGS17V00 Serial: PC0MKUYL	NIC	Intel Corporation, Ethernet Connection (4) I219-LM MAC Address: 54:e1:ad:04:49:39
Motherboard	Vendor: LENOVO Model: 20HGS17V00	Drive 1	Vendor: Samsung Electronics Co Ltd Model: SAMSUNG MZVLW256HEHP-000L7 Serial: S35ENX0J616416 Size: 256.06 GB Drive Media Type: NVME Firmware Version: 4L7QCXB7 EUI64: 002538b671b0b465 Pre-wipe SMART Health Status: PASSED
Processor	Intel(R) Core(TM) i5-7300U CPU @ 2.60GHz	Display Adapter	Vendor: Intel Corporation Model: HD Graphics 620
RAM	8192 MB, Memory banks filled 2/2, DDR4 SODIMM	Multimedia Adapter	Vendor: Intel Corporation Model: Sunrise Point-LP HD Audio
TPM	Found TPM 2.0	USB Ports	USB Ports: 5 USB2 Ports: 0 USB3 Ports: 0
NIC	Intel Corporation, Wireless 8265 / 8275 MAC Address: f8:59:71:49:35:e0		

Hardware Test Overview

Display	Pass	Keyboard	Pass	Memory	Pass	Mouse	Pass
Network	Pass	Processor	Pass	Sound	Pass	Storage	Pass
Web Camera	Pass						

TEXT


log.txt - Notepad
File Edit Format View Help
Operation Info
Action Result: FAILURE
Failure Reason: 1 hardware test(s) had failed results
Software Used: Detalys 1.0.0
Kernel Version: 5.6.13-gentoo-x86_64
Job UUID: 1ED84ABD-AE73-49D7-9EC2-4A85C3969795
Action Start Time: Friday, 16 Oct 2020 at 15:17:45
Action Duration: 00:03:04
Hardware Information
Computer:
Vendor: LENOVO
Model: 20HGS17V00
Serial: PC0MKUYL
Motherboard:
Vendor: LENOVO
Model: 20HGS17V00
Processor: Intel(R) Core(TM) i5-7300U CPU @ 2.60GHz
RAM: 8192 MB, Memory banks filled 2/2, DDR4 SODIMM
TPM: Found TPM 2.0
NIC: Intel Corporation, Wireless 8265 / 8275
MAC Address: f8:59:71:49:35:e0
NIC: Intel Corporation, Ethernet Connection (4) I219-LM
MAC Address: 54:e1:ad:04:49:39
Drive 1:
Vendor: Samsung Electronics Co Ltd
Model: SAMSUNG MZVLW256HEHP-000L7
Serial: S35ENX0J616416
Size: 256.06 GB
Drive Media Type: NVME
Firmware Version: 4L7QCXB7
EUI64: 002538b671b0b465
Pre-wipe SMART Health Status: PASSED
Display Adapter:
Vendor: Intel Corporation
Model: HD Graphics 620
Multimedia Adapter:
Vendor: Intel Corporation
Model: Sunrise Point-LP HD Audio
USB Ports: 5
USB2 Ports: 0
USB3 Ports: 0
Hardware Test Overview
Display: Pass
Keyboard: Pass
Memory: Pass
Mouse: Pass
Network: Fail
Processor: Pass
Sound: Pass
Storage: Pass
Web Camera: Pass
Ln 15, Col 11
80%
Windows (CRLF)
UTF-8

Command Line Options and Parameters

Detalys can be configured on the fly for hardware testing by passing in parameters from the command line using the optional parameters below. In order to access the command line, simply type 'exit' anytime within the GUI and confirm the prompt. When at the command line, you can return to the GUI by typing 'detalys' with the desired options and their parameters.

Command Line Usage:

Example setup: detalys --test-display=all --test-processor=all --test-ethernet=auto

Here is a list of the currently available hardware testing command parameters.

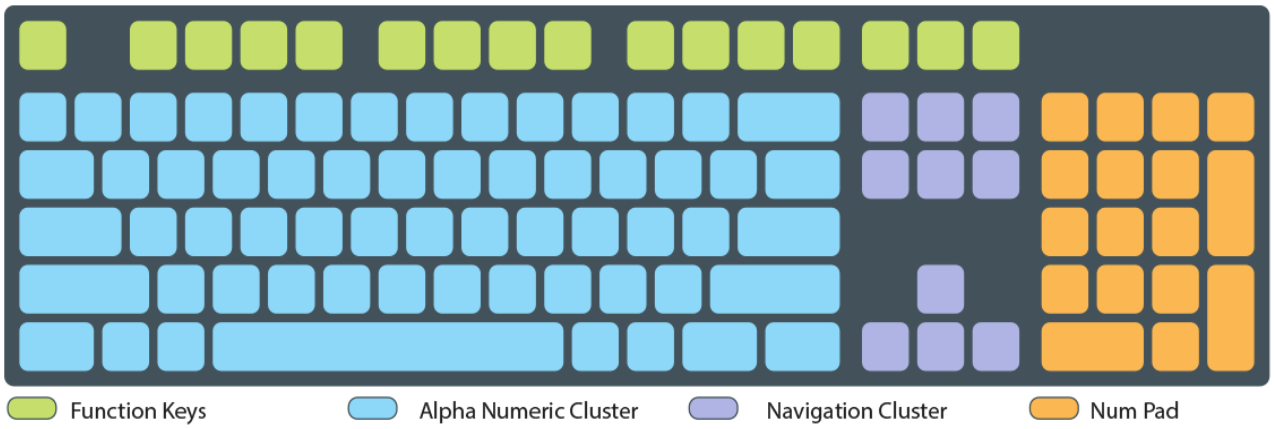
--test-processor=all	Enables the processor test to be performed, testing the device's CPU commands. A comma delimited list can be used instead of 'all' to perform a subset of available tests. Acceptable values are 'mmx', 'sse', 'sse2', 'sse3', 'sse41', 'sse42', 'avx', avx2, fpu, 'prime', and 'aes'. (Example: --test-processor=mmx,sse,avx2,aes)
--test-processor-duration=15	Sets the time duration that the processor tests will run. Accepts any value greater than zero.
--test-storage=all	Enables the storage test to be performed on all selected disks.
--test-storage-duration=15	Sets the time duration that the storage tests will run. Accepts any value greater than zero.
--test-storage-destructive-mode=true	Enables data destructive write operations for the storage test.
--test-display=all	Enables the display test to be performed using all 5 colors.
--test-mouse=all	Enables the mouse test to be performed, testing movement and button functionality.
--test-keyboard=all	Enables the keyboard test to be performed, testing the status lights and keys. A comma delimited list can be used instead of 'all' to perform a subset of available tests. Acceptable values are 'general', and 'lights'. (Example: --test-keyboard=general)
--test-ethernet=auto	Enables the ethernet test to be performed on all available ethernet interfaces.
--test-wireless=auto	Enables the wireless test to be performed on all available and configured wireless interfaces. The 'ssid' option can be provided instead, which will set the test to search for SSID names instead of testing connectivity. A Regex pattern can be added to the search by appending '=' to the option value, followed by a quoted regular expression. (Example: --test-wireless=ssid="whitecanyon.*")
--test-target-ip=8.8.8.8	Designates a destination IP address to use for network testing.

--test-ram=all	Enables the memory test to be performed on the system memory.
--test-ram-duration=15	Sets the time duration that the memory test will run. Accepts any value greater than zero.
--test-webcam=all	Enables the webcam test to be performed on the device's camera.
--test-sound=all	Enables the speaker and microphone test to be performed on the device's audio input/output hardware.
--test-battery=all	Enables the battery test to be performed on the device's battery.
--test-battery-charge-duration=15	Specifies the duration that devices under 50% charge capacity should be charged before the test begins.
--test-battery-discharge-duration=10	Specifies the duration that devices should be tested.
--test-battery-allowable-percent-discharged=15	Specifies the max amount of charge expenditure allowed for the discharge portion of the battery test. Anything over the specified amount will fail the test.
--test-ports=all	Enables the USB ports test to be performed.
--test-disable-cancel=true	Prevents the user from cancelling any hardware test prematurely.

The above-mentioned options and parameters can also be used in the Detalys configuration file. Please contact enterprise-support@whitecanyon.com for assistance in adding these options into a custom build.

Addendum 1 – Custom Keyboard Layouts

Custom keyboard layouts can be specified for the keyboard test by supplying configuration via the '*keyboards.json*' file. This file will specify the name that the layout will be identifiable by, as well as which key regions will be displayed in the test. A keyboard can be broken down into four major areas: The Function Keys row, the Alphanumeric cluster, the Navigation cluster, and the Numpad. All keys associated with these areas can be seen in the following diagram.



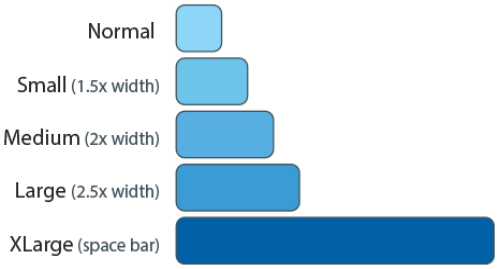
When creating the configuration file for a keyboard layout, the following json attributes need to be given a value.

- "Name": give the keyboard layout a name. This name will be displayed in the keyboard test interface for layout selection.
- "FunctionKeys": specify whether to include the standard function keys row in your keyboard layout. **Note:** If recreating a 60% (laptop) keyboard layout, this row can be more easily specified in the first row of the alphanumeric cluster.
- "NavigationCluster": specify whether to include the standard navigation keys in your keyboard layout.
- "NumPad": specify whether to include the standard 10 key numpad in your keyboard layout.
- "AlphaNumericCluster": specifies the main content of the keyboard. When combined with the function key row, navigation cluster, and numpad, the alphanumeric cluster is typically just 5 rows. When configuring a 60% keyboard layout, a sixth and seventh row can be specified to help contain all the keys in the layout.
- "Misc": if any keys do not fit within the sections previously mentioned, they can be added to this section to be displayed in a minimal manner.

```
"Name": "Lenovo Thinkpad T470s",

"FunctionKeys": true,

"NavigationCluster": true,

"NumPad": true,

"AlphaNumericCluster": {

}

"Misc" : [

]
```

The alphanumeric cluster json object requires the specification of at least five arrays that contain the individual key information for the row. The required attributes for the alphanumeric cluster are: 'Row1', 'Row2', 'Row3', 'Row4', and 'Row5'. 'Row6' and 'Row7' are optional and only needed when specifying laptop keyboard layouts. Each Row is a json array object that will contain a collection of key objects. The key json object has the following attributes that require values to be given.

- "KeyName": the ascii representation for the key object. This should correspond to the Qt::Key code that is specified in the 'KeyCode' attribute.
- "KeyNameUpper": the ascii representation for the secondary value for the key object. This value is what would be accessible when combined with the Shift key.
- "KeyCode": the code name for the key object. Refer to this GUIDE for all supported key codes.
- "KeySize": designates the size of the key when represented visually on the keyboard. The supported sizes are: 'NORMAL', 'SMALL', 'MEDIUM', 'LARGE', 'XLARGE'.




```
{
    "KeyName": "A",
    "KeyNameUpper": "",
    "KeyCode": "Qt::Key_A",
    "KeySize": "NORMAL"
}
```

```
{
    "KeyName": "2",
    "KeyNameUpper": "@",
    "KeyCode": "Qt::Key_2",
    "KeySize": "NORMAL"
},
```

```
{
    "KeyName": "Space",
    "KeyNameUpper": "",
    "KeyCode": "Qt::Key_Space",
    "KeySize": "XLARGE"
},
```

Sample Full Keyboard Layout.
Sample Laptop Keyboard Layout.

Note: Multiple keyboard layouts can be specified in the same '*keyboards.json*' configuration file. Simply use json syntax to create an array of keyboard objects.

Sample Multiple Keyboards Layout.