tl;dr

“CameraProperty” records the properties of each camera on a Windows 10 PC (Surface or non-Surface), and can stream with each media type to verify camera functionality.

HOW

Copy the .\ CameraProperty.zip to a DUT (Device Under Test). Extract to a convenient location, like the Desktop. Run the “Visionary.cmd” script for the default “Stream” test.

CONTENTS

“CameraProperty” consists of 7 files.

- Visionary.ps1 – the script
- Visionary.cmd – a “helper” script to start the default test without any PS windows
- MFCapture*.exe – two 64 bit Win32 executables, one each for ARM64 and X64
- Media.csv, Property.csv – two datafiles
- This readme

PROPERTY TEST

The property test checks for an IR (Infrared “Windows Hello”) camera and up to three RGB cameras, which would correspond to a Surface Pro or Book FFC (“Front Facing Camera”), RFC (“Rear Facing Camera”), and a single EXT (external USB) camera. On a two camera laptop, an external USB camera would appear as the RFC.

Each camera’s KSProperty’s and KSProperty’s Extended are queried and written to a comma delimited csv file in the “\Capture\Property” directory. Each camera’s MediaTypes (including Format, Framerate, Resolution Width and Height) are written to a second csv in the same directory.

This test takes about 2 minutes to complete.

Command line:

```
Visionary -Test Property
```

STREAM TEST

The stream test reads a previously created MediaTypes.csv for the DUT and streams every possible Format, Framerate, and Width/Height for Preview, Capture and Still on each camera.

Each individual attempt is logged, and execution failures logged. The media samples (Capture movie .avi and Still photo .jpg) files are saved. Each media sample is verified for Width/Height, and the movie samples also verified for Framerate and length.

The test run time is a function of the number of cameras and their media types. A typical Surface Pro 5 takes 15 minutes to complete running all media types on all cameras and verification.

Command line:

```
Visionary -Test Stream
```

LOGGING

The “Property” test logs to the “\Capture\Property” directory, creating this if it doesn’t exist. The logs include a master summary, “_CameraPropertyResults.log”, the individual raw camera property interrogation results, and a combined “MEDIA.csv” (all camera media types) and “PROPERTY.csv” (all camera KSProperty, KSPropertyEx).

The “Stream” test logs to the “\Capture\Stream” directory, creating this if it doesn’t exist. The logs include a master summary, “_CameraPropertyResults.log”, the individual log for each camera stream event, all Movie capture samples (*.avi) and all Photo capture samples (*.jpg). If an error is detected in a stream case, the term “_ERROR” is added to that individual log.

Any older results are overwritten.

COMMAND LINE

Visionary.ps1, is a PowerShell script. You do not need ADMIN privilege to run.

You can start the script in multiple ways.

Double-click on the “Visionary.cmd” file.

This starts the script with the default test, “Stream”. You don’t need to open any PowerShell windows or set permissions.

From a PowerShell or command prompt, run “Visionary.ps1”.

This gives more flexibility, but you need to set the PowerShell privileges on the DUT. You can set the script policy manually with this command from an ADMIN PowerShell window, which only needs to be run once.

```
Set-ExecutionPolicy -Execution RemoteSigned -Force
```

Once you have enabled scripts with this command, you can run the SurfCamSanity.ps1 script from either a DOS or PowerShell command prompt.

There are two tests, “Stream” and “Property”. If you just run “Visionary.ps1”, the default “Stream” test is run.

```
Visionary.ps1 – Test Property

Visionary.ps1 – Test Stream
```

There are three additional test flags that can be used for special testing.

The “Camera” flag manually selects an individual camera to test. This is useful for:

- You want to focus on one single device
- There is an EXT external USB camera to test
- You want to run two separate cameras simultaneously for a stress scenario

The “Camera” flag accepts “IR, “FFC”, “RFC”, “EXT” and “ALL” (the default is ALL cameras).

The “Auto” flag repeats the current test until it is manually stopped. There is a 30 second timer between runs to end the test with the keyboard “Q”, <control><Break> or just closing the PowerShell window. Also used for stress scenarios.

The “Auto” flag accepts “Auto” (skip the 30 second timer, run test once), “Repeat” (run until stopped), and “Manual” (default run once with 30 second timer).

The “Datafile” flag is extremely handy for development. The script uses a previously created MediaTypes for the standard recent Surface systems. However, if you have a camera to test that’s not included, or are making changes to the driver and media types, you can just create your own.

First, run the “Property” test to get your device’s cameras and their supported media types.

```
Visionary.ps1 – Test Property
```

Then, use the “Datafile” flag to point to your newly created media types file.

```
Visionary.ps1 – Test Property -Datafile \Capture\Property\NOTSURF.media.csv
```

RGB / IR STRESS TEST

You can stream from any 2 of 3 cameras repeatedly for an effective stress test (streaming from all 3 Surface cameras at one time is not supported).

Open two DOS or PowerShell prompts (does not need ADMIN). CD in each to the test directory. In each one, select a different camera, and use the “Repeat” command. Here are the three possible camera command lines.

```
Visionary –Camera IR –Auto Repeat
```

```
Visionary –Camera FFC –Auto Repeat
```

```
Visionary –Camera RFC –Auto Repeat
```

Each script instance will cycle through all the camera states, then wait 30 seconds for any user request. You can quit at the “Count down” prompt with “Q” or <Control C>.

BONUS

At runtime, the script attempts to identify the DUT as a Surface device or a non-Surface device. Both Surface and non-Surface devices are supported.

Windows 10 X64 and ARM64 are supported. ADMIN privilege is not necessary.