**TH1, TH2, TH3 Scripts Index**

| Thermal 1/1.7 | | Thermal 2 | | | Thermal 3 | |
|---|---|---|---|---|---|---|
| **Service** | **Expert** | **Service** | **Expert** | | **Service** | **Expert** |
| actuator.spt | copyscor.spt | BBsprofile.spt | BPDNoise.spt | ReadPlot Data.spt | BBsprofile.spt | bpd_bode.spt |
| bmcurve_series.spt | creep.spt | BeamBalance.spt | BPDSettle.spt | RemoteUpload.spt | BeamBalance.spt | bpd_bode_fine.spt |
| bmslope_series.spt | headpwr.spt | BPMeasure.spt | DmodSyncvLpos.spt | setsyscomment.spt | contrast_vs_power.spt | currentprofile.spt |
| calcexp.spt | platprof.spt | ContvBP.spt | EdgeGroove3244.spt | shtrglitch.spt | ContvBP.spt | edgedetectfind.spt |
| calibrate_amp.spt | readfocus.spt | ContvHV.spt | EdgeGrooveHydra.spt | TestAll.spt | ContvHV.spt | i_vs_p.spt |
| calibrate_calculator.spt | | FErrorvSR.spt | EdgeGrooveQuantum800.spt | Th2io.spt | NVSDump.spt | ibsprofile.spt |
| contrast.spt | | FocusActLin.spt | EdgeMCE.spt | Th2ver.spt | NVSRestore.spt | pointing_vs_piezo.spt |
| ds_series.spt | | FocusPi.spt | EdgeMPE.spt | Upload.spt | t3_plot_series.spt | rfe_linearity.spt |
| ds_series_constant_step.spt | | ForcePing.spt | EndlessBBSProfile.spt | | TrendsetterSeries.spt | |
| edgesim.spt | | LogHeadInfo.spt | FErrvLPos.spt | | | |
| fieldnvscheck.spt | | LPosvLPos.spt | flsrcheck.spt | | | |
| foctest.spt | | NVSDump.spt | FocLog.spt | | | |
| focusdiag.spt | | NVSRestore.spt | FocusvLPos.spt | | | |
| getcontr.spt | | Parmcheck.spt | FVdelay.spt | | | |
| readdiagmem.spt | | PlotOmnibus.spt | FVdelay_inverted.spt | | | |
| savecfg.spt | | tilt_mce_lg_th2.spt | GetDiagPort.spt | | | |
| sd_hdm_series.spt | | tilt_mce_ts_th2.spt | GraphPing.spt | | | |
| sd_media_series.spt | | tilt_mpe_th2.spt | HeadLogger.spt | | | |
| setdiagmode.spt | | | LogSetup.spt | | | |
| strokcal.spt | | | LposStatsvLpos.spt | | | |
| th1_include.spt | | | M8Measure.spt | | | |
| tilt_mpe_th1x.spt | | | MPEOffsetPlot15.spt | | | |
| tilt_mce_th1x.spt | | | PingvCurr.spt | | | |
| wpower_series.spt | | | ReadLogData.spt | | | |
| wpower_series_constant_step.spt | | | | | | |

## Table of Contents

Thermal 1/1.7 Scripts ..... 3
Thermal 2 and Thermal 3 scripts ..... 6
Thermal 1/1.7 script details ..... 12
Media set up script details ..... 14
Thermal 2 and Thermal 3 script details ..... 18

## Thermal 1/1.7 Scripts

| TH1.X Script Name | Head | Level | Subsys | Description | Results | Notes |
|---|---|---|---|---|---|---|
| actuator.spt | TH1/1.7 | Service | FOC | MCE/MPE. Verifies that the focus actuator moves linearly, with repeatability. | Produces a graph. Here is a sample of a good result. | MPE firmware version must be one that reports `lpos` as part of head sample focus—test that first. |
| bmcurve_series.spt | TH1/1.7 | Service | HDM | MPE-only. Media set up script. | | Options common to all scripts<br>Options specific to certain scripts<br>Using large main-scan shifts<br>Exiting the scripts<br>Troubleshooting<br>Media script MPE patterns |
| bmslope_series.spt | TH1/1.7 | Service | HDM | MPE-only. Media set up script. | | |
| calcexp.spt | TH1/1.7 | Service | | MCE/MPE. Useful exposure calculator. Any parameter in its dialog box may be changed and the script will update the rest of the fields as necessary. Requires Splot v2.68 or later. | Calculator dialog box opens with a set of default values. If the computer running the script is connected to an output device, the script retrieves the current head parameters and uses those values as the calculator defaults. Otherwise, it uses pre-stored approximations. Parameters and calculated values. | |
| calibrate_amp.spt | TH1/1.7 | Service | HEAD | MCE/MPE. First of two scripts needed when replacing a thermal head shutter. Used after disconnecting old shutter. Calibrates shutter amplifier gain using a fixed current from a calibration jig (10-4738A). | The script screen-outputs six values marked G, service-level script, H, I, J, K, and L that must be recorded for use in the second shutter script. | TH1.x Shutter Sensor Replacement Procedure (719-0028A-F) contains full instructions for using this script. |
| calibrate_calculator.spt | TH1/1.7 | Service | HEAD | MCE/MPE. Second of two scripts needed when replacing a thermal head shutter. Used after connecting new shutter to the head. | Calculates the calibration values needed for proper operation of the shutter sensor using four values included with the new shutter and the six values output from calibrate_amp.spt. | TH1.x Shutter Sensor Replacement Procedure (719-0028A-F) contains full instructions for using this script. |
| contrast.spt | TH1/1.7 | Service | HEAD | MCE/MPE. Modified version of getcontr.spt—provides more information. Tests beam pointing actuator, pointing sensors, head alignment, and leakage. | Varies the beam position servo set point between –1000 and +1000 and measures power full and power off at each position. Using this information, the contrast ratio as a function of `bposn` is calculated and displayed. Output and contrast peaks should occur near the same `bposn` and this `bposn` value should be entered and saved in the head NVS. Produces four graphs. | |
| copyscor.spt | TH1/1.7 | Expert | SCORR | MCE/MPE. Copies a `scorr` table from one HDM slot to another. | It produces a graph of the copied `scorr` table. | |
| creep.spt | TH1/1.7 | Expert | | MCE/MPE. Tests light valve creep shutter sensor. | Produces a graph. | |
| ds_series.spt | TH1/1.7 | Service | HDM | MPE-only. Media set up script. | | Options common to all scripts<br>Options specific to certain scripts<br>Using large main-scan shifts |

| TH1.X Script Name | Head | Level | Subsys | Description | Results | Notes |
|---|---|---|---|---|---|---|
| ds_series_constant_step.spt | TH1/1.7 | Service | HDM | MPE-only. Media set up script. | | Exiting the scripts<br>Troubleshooting<br>Media script MPE patterns |
| edgesim.spt | TH1/1.7 | Service | | MCE/MPE. Simulates edge detect by slowly moving the carriage and making head sample focus measurements as it goes. Useful for diagnosing edge detect problems. Determine the edge position manually using the commands: `>> plot edge` and `>> carriage`. The carriage command displays the required position information which is to be entered in the script. | | |
| fieldnvscheck.spt | TH1/1.7 | Service | ALL | MCE/MPE. NVS parameter-checking script. | Generates warning messages for any parameters that are out of range and performs other sanity checks. | |
| foctest.spt | TH1/1.7 | Service | FOC | MCE/MPE. Tests basic focus operation, including setting of focus laser intensity, positioning the head, PSD and electronics, alignment, actuator travel, and nvs settings. | Produces a graph. Sample graphs: good \| bad | Before running this script, position the head over media. |
| focusdiag.spt | TH1/1.7 | Service | FOC | MCE/MPE. Used to diagnose most focus problems and as a post mortem test for focus system problems. | Any time a plot is made, focus details for the entire plot can be stored to a head memory bank. This data can be read and displayed by focusdiag.spt without interfering with regular operations. | More about focusdiag.spt. |
| getcontr.spt | TH1/1.7 | Service | HEAD | MCE/MPE. Tests the beam pointing actuator, pointing sensors, head alignment and leakage. | Produces a graph. Sample graphs: good \| bad | More about getcontr.spt. |
| headpwr.spt | TH1/1.7 | Expert | | MCE/MPE. A sanity test for power and drum speed settings. Calculates exposure on plate in mJ, based on measured power on the nosepiece sensor, drum speed, and drum diameter. Also measures leakage light and reports leakage percentage. | Does not produce a graph. | |
| platprof.spt | TH1/1.7 | Expert | | MCE/MPE. Used to detect loading or plate problems that cause focus learn failures. Measures plate profile at a given carriage position by slowly turning the drum through one rotation and doing head sample focus measurements at each step. Slower than profiling with readfocus.spt, but works with any firmware version. | | |
| readdiagmem.spt | TH1/1.7 | Service | | MCE/MPE. Outputs diagnostic sensor data stored by the head when setdiagmode.spt is used (except for focus information). | Data is graphed in the graphics window, and the axes are labeled to reflect data source(s) and time scale. | More about readdiagmem.spt. |
| readfocus.spt | TH1/1.7 | Expert | FOC | See platprof.spt. | | |
| savecfg.spt | TH1/1.7 | Service | ALL | MCE/MPE. Extracts head parameters and saves them to a file. Useful for logging machine status and is the first script that should be run for maintenance or troubleshooting. | Produces three output files. | Run this script before a firmware upgrade, tests, or other major changes. |
| sd_hdm_series.spt | TH1/1.7 | Service | HDM | MPE-only. Media set up script. | | Options common to all scripts<br>Options specific to certain scripts<br>Using large main-scan shifts<br>Exiting the scripts |
| sd_media_series.spt | TH1/1.7 | Service | HDM | MPE-only. Media set up script. | | |

| TH1.X Script Name | Head | Level | Subsys | Description | Results | Notes |
|---|---|---|---|---|---|---|
| | | | | | | Troubleshooting<br>Media script MPE patterns |
| setdiagmode.spt | TH1/1.7 | Service | | MCE/MPE. Determines which data is stored using the head diagnostic capabilities. This script will also start and stop data collection one time. If you run this script a second time, the original data will be overwritten. | This script does not read out any data—it is a tool for setting the values of head parameters `diag0`, `diag1`, and `diag5`, which control the data collection mode. | More about setdiagmode.spt. |
| strokcal.spt | TH1/1.7 | Service | | MCE/MPE. Tests basic laser and LV operation. Use for stroke cal problems, or in case of banding. | Results indicate if the laser profile is square enough and if there are any stuck pixels.<br>Sample graphs: good \| bad | More about strokcal.spt. |
| th1_include.spt | TH1/1.7 | Service | | Not a stand-alone script. Contains common utilities, which the other field scripts call. | | Must be stored in the same directory as the other field scripts, in order for the field scripts to function properly. |
| tilt_mpe_th1x.spt / tilt_mce_th1x.spt | TH1/1.7 | Service | | MCE/MPE. These scripts are used to help determine how much the head is tilted on the carriage. They should be used in conjunction with a lenticular tilt tool (690-00225A). | When used with the default settings, the script will image thirteen 1” strips on a plate. The strips have a geometric correction adjustment to simulate a varying amount of tilt, from –0.75 to +0.75. After it runs, the script will screen–output a listing of the strips:<br>Strip: 1 Tilt: -0.750 Clicks: 7 down<br>Strip: 2 Tilt: -0.625 Clicks: 6 down<br>Strip: 3 Tilt: -0.375<br>By using the tilt tool to determine the strip with the least tilt, the amount of adjustment needed (if any) can be determined from the output listing. | See *Adjusting Tilt on a Thermal Head* for instructions on using this script. |
| wpower_series.spt | TH1/1.7 | Service | HDM | MPE-only. Media set up script. | | Options common to all scripts<br>Options specific to certain scripts<br>Using large main-scan shifts<br>Exiting the scripts<br>Troubleshooting<br>Media script MPE patterns |
| wpower_series_constant_step.spt | TH1/1.7 | Service | HDM | MPE-only. Media set up script. | | |

## Thermal 2 and Thermal 3 scripts

TH2 and TH3 Service Scripts | TH2 and TH3 Expert Scripts

The write laser must be on while executing any of the Thermal 2 Splot scripts. To turn it on, do one of the following:

- Turn on the laser by hand before running the scripts. Turn on the beam pointing servo and set up the light valve driver to local mode. For example, to turn the laser on to 20A, then issue the commands:
  - th2> bpd mode on
  - th2> lvd mode local
  - th2> ldd mode current 20

  After the script has terminated, turn off the laser with the command ldd mode off.
- Configure the scripts to automatically turn on the laser by adding the following lines to the LocalJig_Include.spt file:
  - #define TII_STATION_FIELD
  - #define LASER_TO 20

The second line is the timeout, in seconds, for turning on the laser. This value may vary.

### TH2 and TH3 Service scripts

| TH2/TH3 Script Name | Head | Level | Subsys | Description | Results | Notes |
|---|---|---|---|---|---|---|
| BBsprofile.spt | TH2, TH3 | Service | BBS | Measures the profile of the write laser beam with the beam balance sensor (BBS). The script output helps show how difficult the write laser beam will be to balance. | BBSprofile graphs<br>BBSProfile data<br>Data interpretation | |
| BeamBalance.spt | TH2, TH3 | Service | BBS | Executes the firmware command `bbs balance` and displays the results graphically. The script output helps show how much buffer is left for future aging of the write laser and light valve. | BeamBalance graphs<br>BeamBalance data<br>Data interpretation | |
| BPMeasure.spt | TH2 | Service | BPD | Measures the BP sensor readings as the laser current is increased to the nominal laser current setting. | Produces graphs and a data file. | More about BPMeasure.spt |
| contrast_vs_power.spt | TH3 | Service | BBS | Plots BBS contrast vs power at 10, 20, 30, 40, and 45. | Produces two graphs: contrast on/off vs power and contrast ratio vs power. | |
| ContvBP.spt | TH2, TH3 | Service | BBS | Measures the contrast ratio (off light over on light), using the beam balance sensor, as the beam position offset is changed. | ContvBP graphs<br>ContvBP data<br>Data interpretation | |
| ContvHV.spt | TH2, TH3 | Service | BBS | Measures the contrast ratio (off light over on light), using the beam balance sensor, as the high voltage trim is changed. | ContvHV graphs<br>ContvHV data<br>Data interpretation | |
| FErrorvSR.spt | TH2 | Service | FOC<br>MED | Measures graph focus error versus surface reflectivity and sum versus surface reflectivity. The receiver gain (`foc rcvrgain`) setting at the start of execution is used. | FErrorvSR graphs<br>FErrorvSR data<br>Data interpretation | Instructions |
| FocusActLin.spt | TH2 | Service | FOC | Graphs lens position versus focus actuator current. Used to determine range of motion of actuator and find the point of | FocusActLin graphs<br>FocusActLin data | For version C and later boards, use the ForcePing.spt scripts to set up the |

| TH2/TH3 Script Name | Head | Level | Subsys | Description | Results | Notes |
|---|---|---|---|---|---|---|
| | | | | inflection for parameterization when calibrating the focus servo on a version B HC board. | Data interpretation | focus servo parameters. |
| FocusPi.spt | TH2 | Service | FOC | Graphs the focus laser power versus drive current. Used to check for focus laser diode damage or failure. | FocusPi graphs<br>FocusPi data<br>Data interpretation<br>Example of a normal laser.<br>Example of a damaged laser | . |
| ForcePing.spt | TH2 | Service | FOC | For instructions on setting up the lookup table, see: How to Set up the look up table using forceping.spt . | | |
| LogHeadInfo.spt | TH2 | Service | ALL | Copies a snapshot of the head status to the results folder. | Records NVS settings and head information in the log file loghead.dat which can be viewed with any text editor, including NotePad. | |
| LPosvLPos.spt | TH2 | Service | FOC | Graphs commanded lens position vs actual lens position. Used to assess position servo performance, stability, and range of motion. | LPosvLPos graphs<br>LPosvLPos data<br>Data interpretation | |
| NVSDump.spt | TH2, TH3 | Service | ALL | Creates a backup of head NVS parameters. | Records the current NVS setting. The readable output file, PRMParms.txt, contains all NVS settings. | |
| NVSRestore.spt | TH2, TH3 | Service | ALL | Restores head NVS parameters from the NVSDump data file. | Searches the results directory for most recent NVSDump results. If this is not the desired NVS result, you can enter a different path and filename. Script reports on success of the update. | |
| PlotOmnibus.spt | TH2 | Service | MED | MPE only. Allows you to do a power, curve, beam position, offset, or slope series. Using a dialog box, you can choose to enter a firmware command, do a plot, or quit. | Produces no graphs or data. | |
| t3_plot_series.spt | TH3 | Service | | TH3 series plot utility that optimizes Th3 focus, slope, curve, Surface depth, power. | This script creates a plot series. | |
| tilt_mce_lq_th2.spt | TH2 | Service | | MCE/MPE. Used to help determine how much the head is tilted on the carriage. These scripts should be used in conjunction with a lenticular tilt tool (690-00225A). | When used with the default settings, the script will image thirteen 1” strips on a plate. The strips have a geometric correction adjustment to simulate a varying amount of tilt, from –0.75 to +0.75. After it runs, the script will screen-output a listing of the strips:<br>Strip: 1 Tilt: -0.750 Clicks: 7 down<br>Strip: 2 Tilt: -0.625 Clicks: 6 down<br>Strip: 3 Tilt: -0.375<br>By using the tilt tool to determine the strip with the least tilt, the amount of adjustment needed (if any) can be determined from the output listing. | See *Adjusting Tilt on a Thermal Head* for instructions on using this script. |
| tilt_mce_ts_th2.spt | TH2 | Service | | | | |
| tilt_mpe_th2.spt | TH2 | Service | | | | |
| TrendsetterSeries.spt | TH2 | Service | MED | MCE only. Used to perform series plots on Trendsetter devices. Note that this is now superseded by TPA which is available for all | This script creates a plot series. | |

| TH2/TH3 Script Name | Head | Level | Subsys | Description | Results | Notes |
|---|---|---|---|---|---|---|
| | | | | MCE-based TS devices. | | |

### TH2 and TH3 Expert scripts

| TH2/TH3 Script Name | Head | Level | Subsys | Description | Results | Notes |
|---|---|---|---|---|---|---|
| bpd_bode.spt | TH3 | Expert | BPD | Creates and displays BPD bode plots. | Produces two graphs. Example of a good result. | |
| bpd_bode_fine.spt | TH3 | Expert | BPD | Creates and displays BPD bode plots. | Produces two graphs. | |
| BPDNoise.spt | TH2 | Expert | BPD | Used to measure noise on beam pointing (BP) sensor. If noise is too high, the BP servo will not position the beam accurately. | Example graphs. | More about BPDNoise.spt. |
| BPDSettle.spt | TH2 | Expert | BPD | Used to show the performance of the beam pointing (BP) servo. If the BP servo performs poorly, there may be artifacts on the plate. | Example graphs.. | More about BPDSettle.spt. |
| currentprofile.spt | TH3 | Expert | BBS | Graphs the full beam width of beam profile at level 0 and at level 255. Note that bbs profile changes the gain and offset so that the curves are not trivially comparable. | Produces a graph and data. | |
| DmodSyncvLpos.spt | TH2 | Expert | FOC | Used to ensure that the range finder electronics is performing correctly. This script must be run with the head over a plate, with the reflectivity and receiver gain set correctly. | Typical output: Example Graph 1. All of the lines should come together around `DModSync` = 20 to 30, and the difference should be between –100 and +100. Ideally, the maximums and minimums around `DModSync` = 70 should align. The optimal value of `DModsync` occurs at this location. The FVDelay.spt script should be used to accurately set `DModSync`. | |
| edgedetectfind.spt | TH3 | Expert | | Used to optimize the edge detection algorithms. | Records SUM measurements with carriage and drum positions in a script result file, so they can be interpreted for the best edge strip position (`edge drumpos`). | More about edgedetectfind.spt. |
| EdgeGroove3244.spt | TH2 | Expert | | Used to determine the best drum position for the plate edge detect sequence. The 3244 version requires a plate to be loaded and assumes the host controller is an MPE. | Example graphs | More about EdgeGroove3244.spt |
| EdgeGrooveHydra.spt | TH2 | Expert | | Used to determine best drum position for the plate edge detect sequence. The script requires a plate to be loaded and assumes the host controller is an MCE and the drum is similar to a 3244 drum. | Example graphs | More about EdgeGrooveHydra.spt |
| EdgeGrooveQuantum800.spt | TH2 | Expert | | Used to determine the best drum position for the plate edge detect sequence on a Lotem 800 Quantum. | Example graph | About EdgeGrooveQuantum800.spt |
| EdgeMCE.spt | TH2 | Expert | FOC | Used to map focus SUM and Error signals at edge of plate over a range of carriage positions, but at a fixed drum position. | | It can be desirable to observe the edge of the plate for excessive plate curl and or SUM signal roll off at the edge.<br>When to run the script \| Instructions |
| EdgeMPE.spt | TH2 | Expert | FOC | Used to map focus SUM and Error signals at edge of plate over a range of carriage positions, but at a fixed drum position. This | | It can be desirable to observe the edge of the plate for excessive curl and or |

| TH2/TH3 Script Name | Head | Level | Subsys | Description | Results | Notes |
|---|---|---|---|---|---|---|
| | | | | version assumes an MPE is hosting the head. | | SUM signal roll off at the edge.<br>When to run the script \| Instructions |
| EndlessBBSProfile.spt | TH2 | Expert | BBS | Used to measure the consistency of the beam profile. This script can help find intermittent bad pixels. | Performs continuous BBS PROFILE and displays the data in the Splot graph window. Simple statistics are also performed on the data. The graph displays the first, most recent, maximum, minimum, and average beam profile. Maximum, minimum, and average are calculated for each pixel. | More about EndlessBBSProfile.spt |
| FErrvLPos.spt | TH2 | Expert | FOC | Graphs focus error versus lens position. It is used to check the linearity of the range finder and to calibrate errorcal. | A graph, FErrvLPosn.ps, shows error reading, in raw units, versus lens position, in microns. The output file, FErrvLPos_data.txt, has two columns: lpos in microns, and error reading. | The graph should be fairly straight and the quality of fit, r, specified in the bottom left corner of the graph, should be at least 0.9995. The value of errorcal, shown in the lower left corner of the graph, should be 155, ±30. |
| flsrcheck.spt | TH2 | Expert | MED | Checks the focus laser, head electronics, and head firmware for mutual compatibility. Heads in the field should not experience compatibility problems unless someone has downgraded the head to obsolete head firmware. | Data file | |
| FocLog.spt | TH2 | Expert | FOC | Interactive script, used to extract and display real-time head log data, or to read and display previously extracted head logs data—you can plot graphs repeatedly. | | If extracting head data, it is assumed the head log has been acquired and the data is in head memory. Typically, this script is used in cases where data is drum position synchronous. ReadLogData.spt is usually used for non-drum position synchronous data.<br>When to run this script \| Instructions |
| FocusvLPos.spt | TH2 | Expert | FOC | Plots focus sum, focus difference, and focus difference divided by focus sum vs lens position. Surface reflectivity and receiver gain settings at the start of execution are used. These two values, plus dmodsync and LPosCal, are shown at the bottom of each graph. | FocusvLPos graphs<br>FocusvLPos data<br>Data interpretation | Instructions<br>**Note**: A plate needs to be loaded for this script to work and the head should be installed with an error of 0. |
| FVdelay.spt | TH2 | Expert | FOC | Used to determine the optimal focus receiver demodulator phase. | graph \| data file | Instructions |
| FVdelay_inverted.spt | TH2 | Expert | FOC | Used to determine the optimal focus receiver demodulator phase. FVdelay_inverted only works 180 degrees out of phase vs FVDelay.spt and gives access to the other 180 degrees of phase. | graph \| data file | Out of date script—only useful on version B Head Controller (HC) boards and some version C boards with older firmware. Instructions |
| GetDiagPort.spt | TH2 | Expert | | Used to acquire host/head serial port and switch to be the diagnostic port in cases where the break-in hardware card is used. The script takes care of handshaking requirements so the head will know to switch the port as it comes out of reset.<br>Note that this is only used in production. The field will not have a head breakout card available to communicate with the head. | | Use this script when breaking into the head using the break-in card, any time you want to check I/O performance. |

| TH2/TH3 Script Name | Head | Level | Subsys | Description | Results | Notes |
|---|---|---|---|---|---|---|
| GraphPing.spt | TH2 | Expert | | Used to show an autofocus current step response. The script pings the actuator at 100Hz (sends a '`foc cal svf start_cur end_cur 100`' firmware command). | Produces a graph and a data file. | |
| HeadLogger.spt | TH2 | Expert | | Used to log TH2 statistics. | | More about HeadLogger.spt |
| i_vs_p.spt | TH3 | Expert | BBS | Compares laser currents vs beam power. | Produces one graph and a data file.<br>• Current_vs_power | |
| ibsprofile.spt | TH3 | Expert | BBS | Graphs the full beam width of beam profile at level 0 and at level 255. Note that ibs profile changes the gain and offset so that the curves are not trivially comparable. | Produces three graphs and a data file. Sample graphs:<br>• Ratio<br>• IBSProfileAll<br>• IBSProfileAllAdj | |
| LogSetup.spt | TH2 | Expert | | Used to set up the TH2 data logger. The data logger can be used to collect real time information about various actuators and sensors in the head. | | More about LogSetup.spt |
| LposStatsvLpos.spt | TH2 | Expert | FOC | Used to measure the autofocus positioning stability. | Produces two graphs and a data file. | |
| M8Measure.spt | TH2 | Expert | LDD | Measures the M8 power sensor readings as the laser current is increased up to the nominal laser current setting. | Produces a graph and a data file. | See also Data interpretation. |
| MPEOffsetPlot15.spt | TH2 | Expert | FOC | Used to determine the best offset setting for a particular media. This script does a plot 15 (1 pixel checkerboard). | Produces no graphs or data. See the media setup documentation to interpret the resulting plot. | Before you run the script.<br>When you run the script. |
| Parmcheck.spt | TH2 | Expert | ALL | Used to check that variables for TH2 heads are properly set up. The script parses the second revision NVSDump files "PRMParms.txt". You should use an up-to-date NVSDump file. | HTML file of variables \| LUT graphs | When and how to run this script. |
| PingvCurr.spt | TH2 | Expert | FOC | Used to measure the autofocus characteristics for various actuator currents. | Produces a graph and a data file. | More about PingvCurr.spt |
| pointing_vs_piezo.spt | TH3 | Expert | LDD | Plots DBP sensor diff and sum vs beam pointing. Fit to determine `fwhm` on sensor. | Creates six graphs: A and B quadrants, A and B PSD, PSD (position vs voltage, and LDD (power vs voltage) | |
| ReadLogData.spt | TH2 | Expert | | Used to extract and display real-time head log data or read and display previously extracted head log data. Use this interactive script to extract and display head log data in the form of simple sample vs sample time—you can plot graphs repeatedly. | | More about ReadLogData.spt |
| ReadPlot Data.spt | TH2 | Expert | | Used to extract and display head real-time head log data or read and display previously extracted head log data. | Example graphs and dialog box. | More about ReadPlotData.spt |
| RemoteUpload.spt | TH2 | Expert | | Used to upload a .bin firmware file to a remote head. At the prompt, enter the full path and filename of the .bin firmware file on the remote machine. | Produces no graphs or data. | This script will not work if the firmware file is on the local machine. Splot must be communicating through the high speed diagnostic port to use this script. |

| TH2/TH3 Script Name | Head | Level | Subsys | Description | Results | Notes |
|---|---|---|---|---|---|---|
| rfe_linearity.spt | TH3 | Expert | | Plots RFE receiver gain linearity. | Produces a graph. Examples of a good result. | |
| setsyscomment.spt | TH2 | Expert | SYS | MCE/MPE. Updates the `sys comment` field in TH2 heads with a standardized description based on the part number chosen from a drop-down menu. | | More about setsyscomment.spt. |
| shtrglitch.spt | TH2 | Expert | BBS | Checks for noise on the shutter. | This produces two graphs, one for when the shutter opens, one for when it closes. If there is too much noise the graphs indicate failure (sentence at the bottom right of the graph). | |
| TestAll.spt | TH2 | Expert | ALL | Runs the Splot prescan on all scripts in the current directory. Requires Splot 3.90 or later. | The script saves no output. Any errors will be reported in the errlist window. | |
| Th2io.spt | TH2 | Expert | | A system diagnostic tool used to measure and rate the I/O performance from the head back to system operating the script. | Example script output. | Instructions |
| Th2ver.spt | TH2 | Expert | | Used to check versions of include files needed for TH2 script operation. Also lists some installation configuration settings. You do not have to be connected to a device to run this script. Use this script any time you want to check versions or if you are having problems and suspect an incorrect version of an include file. | | Instructions |
| Upload.spt | TH2 | Expert | | Uploads a .bin firmware file to the head. | Produces no graphs or data. | Splot must be communicating through the high speed diagnostic port to use this script. This script does not work with a remote hookup. Also, you should only specify the filename. The path is specified in the LocalJig_Include.spt file, in the following line: #define DEFAULTBUNDLEPATH "W:\\firmware\\hc\\". If there is an error, there will be an error message in the terminal.log file. |

# Thermal 1/1.7 script details

**calcexp.spt**

Parameters and calculated values:

- Nose piece power in W
- Drum diameter in mm
- Drum speed in RPM
- Effective swath width in pixels
- Head DPI
- Exposure in mJ/cm2
- Surface velocity in mm/sec
- Sub scan speed in mm/sec
- Imaging time in sec
- Stroke rate in KHz
- Data rate in MHz

**contrast.spt**

Produces four graphs. The graphs are displayed in succession. The first three graphs display power full, power off, and contrast individually. Note that the scale is altered for the first three graphs.
This allows you to see more detail in each of the lines plotted. The fourth graph displays all three lines together, as in getcontr.spt. The fourth graph should be consulted before determining beam position.

**fieldnvscheck.spt**

Generates warning messages for any parameters that are out of range and performs other sanity checks. Errors or warnings generated may not actually reflect a problem.
For example, a parameter may have to be adjusted outside the normal range to allow operation of the head. However, the script provides a good starting point to investigate possible problems.
You must not adjust parameters into range to get rid of the warnings if you don’t understand what the parameter does or the implications of the adjustment.

**focusdiag.spt**

To set the head to store focus information, use setdiagmode.spt to configure the upper and lower banks of diagnostic memory to Focus Data.
You can store focus information or other head parameters with setdiagmode.spt, but not both.
Focus Data mode allows focus information to be stored and turns off other data collection. Previously stored information will be over-written when the diag mode is changed—if you have stored humidity (or other parameters) for a long period, you may want to output the data before changing settings.
Focusdiag.spt only reads out previously stored focus data—it does not configure the head to collect data. When the script runs, you will be asked for a path to a data storage location as well as the start and stop swaths. Reading out an entire plate-worth of focus diagnostic information can take 15 minutes or more. You should restrict the readout to the sub-scan area of interest. To do this, select a start and stop swath in the region of interest. By default, start and stop values correspond to the first and last swath from the last plot. When the script has finished extracting all the data from the head, it will display a movie of the focus performance. One to four swaths (user configurable) are displayed at any one time and the script will play them all from start to stop. There are dialog buttons to pause, quit, step forward, back, etc.
The movie will terminate on the last swath. To view the focus performance again it is not necessary to re-read the data from the head. Instead, by answering "n" to the "Collect data from head" question at the start of script execution the data is read from the local data file (at the user-specified location) created the first time the script was run. This is much faster—less than five minutes in most cases.

**getcontr.spt**

This script varies the beam position servo set point between –1000 and +1000 and measures power full and power off at each position. From this information the contrast ratio as a function of `bposn` is calculated and displayed. The peak in the output and the peak in the contrast should occur at near the same `bposn` and this value of `bposn` should be entered and saved in the head NVS.

**readdiagmem.spt**

When this script is run you will be asked for a file name to store the results to on your local computer. A simple choice is something like "c:\temp\data".
Then, if the number of data points stored is large, a dialog will allow entry of a reduced number of data points to read out. Reading out data from the head is slow and reading the entire memory contents may take fifteen minutes or more if both the lower and upper banks are full. Once data has been read out, it is graphed in the graphics window, and the axes are labeled to reflect the data source(s) and the time scale. To view the data again it is not necessary to read the data from the head again. Instead, by answering "n" to the "Collect data from head" question at the start of script execution the data is read from the local data file (at the user specified location) that was created the first time the script was run. This is considerably faster, taking less than five minutes in most cases.

**savecfg.spt**

Produces three output files. Previous versions of files produced by this script are retained and renamed as LD0046.sp1, LD0046.sp2, etc. These files are put in a subdirectory from the current directory named TH0046 where the number is the head serial number 46 in this example. Savecfg.spt generates an output file of the form TH0046.cfg where the number in the name is the head serial number.
It also produces a strokcal.pdf file with the stroke cal results displayed graphically. The script does a reset head at the end, so be sure to save any changed head parameters first.
This script records every nvs parameter relevant to the head as well as the scorr tables for all defined media. It captures the current state of the head and produces a new script called ldthxxxx.spt, which can be run to return the head to this state.

# Media set up script details

**Options common to all scripts**

Each program starts with a menu of options. Options that appear on all scripts are listed below.

- **Media Type** allows you to pick which media type to use during plotting. If you add a new media or change which hdm or media slot a media is kept in, you must refresh the settings by clicking Refresh Media.
- **Patterns** allows you to choose the patterns you wish to plot. You can plot more than one pattern by writing a list of patterns separated by a comma. For a list of available patterns, type "list pattern" in the terminal in Splot.
- **Strip Spacing** is the spacing between plots in tenths of an inch.
- **Sub-Scan Offset** is the offset along the drum of the first plot in mils (`mode n sssa`).
- **Main-Scan Offset** is the offset around the drum of the plots in mils (equivalent to `mode n mssa`).
- **Strip Width** is the width of the plots in tenths of an inch.
- **Strip Height** is the height of the plots in tenths of an inch.
- **Use Edge Detect** determines whether or not an edge detect is done before writing the first plot.
- **Reload Defaults** reloads the settings that were set when you started the script.
- **Start Series** starts running the script with the values you have set. It also saves those settings and will start up with those settings the next time you run that script.
- **Refresh Media** queries the Trendsetter to determine what media have been installed on the device, then lists them in the Media Type box. If you add a new media or change which hdm or media slot a media is kept in, you must refresh the settings by clicking this button.
- **Quit** stops the script without saving any of changes you made to the script parameters.

**Options specific to certain scripts**

The wpower_series.spt and ds_series.spt scripts ask you to set a series of write laser powers or drum speed values for their series. Input a series of the wpowers or drum speeds that you want to plot with, separated by commas.

The sd_series.spt, bmcurve_series.spt and bmslope_series.spt scripts ask for an initial value, step value and number of plots. These are the same options that you would have using a firmware series.

If you want to output a slope series from -9 to +9 with steps of 1, set the initial slope to -9, the slope step to 1 and the number of plots to 19.

**Using large main-scan shifts**

The total main scan shift is the sum of `gc mss`, `gc mssa` and `mode n mssa`. If the total main-scan shift is larger than 3000, the script will warn you that you have to have plot shift large on, but will continue running whether or not you have done it. If you want to set large main scan shifts, make sure that you turn on large plot shifts by typing: `>>plot shift large`, and turn off range-checking by typing `>>mask range` in the terminal log. Older versions of MPE firmware will not support large main scan shifts. To go back to normal main scan shift size, type: `>>plot shift normal` and `>>mask range`.

**Exiting the scripts**

When the script ends, it will reset all of the parameters it has changed to whatever they were before the script started running. If you try to stop the script too many times (by using Exec > Stop Execution more than once, for example), you will interrupt the script before it has finished doing this, and your firmware parameters may or may not be reset to their initial values. If you need to exit the script quickly, make sure that you check that the parameters did get reset to their default values. If the script exits with an error, a message box will come up telling you what went wrong. If the script exits normally, it will print "*** Script completed successfully ***" near the end of its output. **NOTE**: *The scripts do not do a stroke cal after resetting the parameters. You will need to do this manually*.

**Troubleshooting**

If the media and hdm slots don't match up, click on the Refresh Media button to query the *Trendsetter*'s media and HDM slots again.

If the script tells you that you need a newer version of Splot, then install the version supplied on the CD-ROM. If this doesn't work, look at the output window and try to figure out what command the script is having trouble with. If the script is timing out, it may be a hardware problem, or the firmware may have changed its replies so that the scripts are no longer compatible.

**Media script MPE patterns**

These are the numbers that you input into the Pattern box when running the scripts. Unless otherwise noted, all of the regular AM screens are Euclidean at 212 lpi and 45°.

| Pattern | Description | Comments |
|---|---|---|
| 0 | Universal pattern | |
| 1 | Microband pattern | |
| 2 | Microband2 pattern | |
| 3 | Exposure series | Black pattern |
| 4 | Exposure series | Screen pattern |
| 5 | Blank pattern—all zeroes | 100% screen for positive media (invert=1) |
| 6 | DPE Test 0 | Universal DPE test pattern (all DPE patterns) |
| 7 | 3 pixel horizontal bars | |
| 8 | Focus Series | |
| 9 | Microband pattern | 150 lpi |
| 10 | Step Wedge | |
| 11 | Paul’s 2 vbar | |
| 12 | Solid pattern—all ones | 100% screen for a negative media (invert=0) |
| 13 | 3 pixel vertical bars | |
| 15 | 1 pixel checkerboard | Equivalent to a square 50% at 1700 lpi, 45°. Very hard to image. |
| 22 | 2 pixel checkerboard | Equivalent to a square 50% at 849 lpi, 45°. Comparable to Staccato 10 and 20.1. |
| 33 | 3 pixel checkerboard | Equivalent to a square 50% at 566 lpi, 45°. Comparable to Staccato 20. |
| 44 | 4 pixel checkerboard | Equivalent to a square 50% at 424 lpi, 45°. |
| 55 | 5 pixel checkerboard | Equivalent to a square 50% at 339 lpi, 45°. |
| 66 | 6 pixel checkerboard | Equivalent to a square 50% at 283 lpi, 45°. |
| 77 | 7 pixel checkerboard | Equivalent to a square 50% at 242 lpi, 45°. |
| 88 | 8 pixel checkerboard | Equivalent to a square 50% at 212 lpi, 45°. |
| 99 | 1x1 to 8x8 checkerboards | |
| 16 | Microband pattern, 283 lpi | |
| 18 | 3 pixel diagonal bars | |
| 19 | Microband6 pattern | Horizontal |
| 30 | 30% screen | Euclidian screen at 212 lpi, 45° |
| 50 | 50% screen | Euclidian screen at 212 lpi. Equivalent to an 8x8x checkerboard (pattern 88). |
| 80 | 80% screen | Euclidian screen at 212 lpi, 45° |
| 93 | 93% screen | Euclidian screen at 212 lpi, 45° |
| 97 | 97% screen | Euclidian screen at 212 lpi, 45° |
| 100 | Crosshair pattern for GC | |
| 101 | DPE Test 1 | Alternating 1’s and 0’s in sscan/mscan |
| 102 | DPE Test 2 | Same as DPE Test 1, but different signature |
| 103 | DPE Test 3 | Walking 1’s in mscan |

| | | |
|---|---|---|
| 104 | DPE Test 4 | Walking 0’s in mscan |
| 105 | DPE Test 5 | Walking 1’s in mscan |
| 106 | DPE Test 6 | Walking 0’s in mscan |
| 107 | DPE Test 7 | 0.2 inch horizontal bars |
| 110 | 10% screen | Euclidian screen at 212 lpi, 45° |
| 111 | 1 pixel vertical bars | |
| 112 | Quarter-wide plot 1 | |
| 113 | Digital Masking Test 1 | Checkerboard |
| 114 | Dashed crosshairs | For GC matching |

**setdiagmode.spt**

Using this script, you can configure the head to collect and store data from any two sensors (similar to a 2-channel digital storage scope).
A limitation is the low data transfer rate out of the head. You can collect a lot of data quickly, but it can take many minutes to read it out. When you start the script,
a button menu allows selection of the first diagnostic memory mode. After selecting both data sources, you will be shown how long it will take to fill the buffer—you can enter a divisor to change this.

**Diagnostic memory mode choices**:

1. Focus Data: use for upper and lower banks, to configure for use with focusdiag.spt
2. Write Laser: write laser power sensor reading
3. BP0 Data: BP0 sensor reading
4. BP1 Data: BP1 sensor reading
5. Nosepiece: shutter sensor power reading
6. Diode Temp: laser diode temp reading
7. Ambient Temp: ambient temp reading
8. Case Temp: case temp reading
9. Humidity: humidity reading
10. LV Temp
11. Focus Sum: focus sum signal
12. Focus Error: focus error signal
13. Diag0 Data: leads to additional menu of choices, see below
14. Diag1 Data: leads to menu of additional choices, see below
15. Lens Position: lens position reading

Each sensor has its own characteristic data collection rate. To match the time scale for both data streams, the first channel (lower bank) sets the overall time scale, and the data for the second channel
(upper bank) can be adjusted as necessary. Total available sample space is 30K for each channel. After 30K has been collected, data acquisition stops. The time to fill the sample memory depends
on the characteristic data rate of the channel 1 source—it may be as short as 6 seconds. To collect data for a longer time period or reduce the number of samples to be read out of the head,
you can set a divisor (the `diag5` parameter). For example, if a divisor is set to 10, only every tenth sample is stored, and it takes 60 seconds to fill the memory instead of 6 seconds with a divisor of 1.
Setting the divisor to the largest value of 32767 allows collection of environment information such as humidity for a period of months. Setting the divisor to 0 turns off data collection for the readdiagmem.spt script.
If the head is configured to collect information for the focusdiag.spt script, it will still collect this data while plotting, even if the divisor (`diag5`) is 0.
It is important to know what starts and stops data acquisition. This also depends on what input was selected for the first channel. The first channel is the master and the second is the slave.
The second channel starts and stops at the same time as the first. If the channel 1 data source is the write laser power, the BP sensors, or the shutter sensor, data collection starts and stops
every time the laser is turned on or off. If the channel 1 data source is the focus sum, focus error, or lens position, data collection starts and stops when the focus is turned on and off.

For the environment sensors or the `diag0` and `diag1` outputs, data collection starts immediately after the script sets up the data acquisition mode, and stops when the acquisition mode is set back to “Focus Data”. Note that every time data collection is stopped and restarted, any previous unread data is lost—only the most recently collected data is available. Diag0 Data and Diag1 Data modes allow you to configure the head’s original diagnostic output capabilities. The head has a diagnostic port to which you can connect a cable and view signals from the head. Instead of connecting a cable, you can loop the output back into the head and store these values. Configuring these data sources is separate from configuring what to store in the head diagnostic memory and is handled by the script if these data sources are chosen.

Another menu of output parameters is available, with some parameter duplication:

| **Diag0 Data mode options** | **Diag1 Data mode options** |
|---|---|
| 1. Focus Error | 1. Focus Sum |
| 2. Lens Position Error | 2. Focus Diff |
| 3. Lens Position | 3. Focus Actuator Current |
| 4. Focus State | 4. Focus State |
| 5. Focus Control | 5. Focus Control |
| 6. Laser Power Error | 6. Laser Current |
| 7. Laser Current | 7. Laser Power |
| 8. BP Error | 8. BP Current |
| 9. BP Current | 9. BP Sum |
| 10. Stroke Cal Pulse Heights | 10. Focus Error |
| 11. Stroke Cal Total Off | 11. Stroke Cal Pulse Heights |
| | 12. Stroke Cal Total On |

There are 3 script configuration scenarios. Use the first if focus information is to be analyzed using focusdiag.spt. The upper and lower diagnostic memory banks must be set to “Focus Data” (`diag0` and `diag1` set to 0). Readdiagmem.spt will not work in this memory configuration. Use the second scenario if the other head parameters will be analyzed with readdiagmem.spt. In this case, configure the banks to the two parameters desired (other than “Focus Data”). Use the third scenario when analyzing only one head parameter with readdiagmem.spt. In this case, configure the first channel (lower bank) to the desired parameter and the second channel (upper bank) to “Focus Data”. Only the desired parameter will be displayed by readdiagmem.spt.

The script displays two option buttons: COLLECT MANUALLY and COLLECT USING SCRIPT. COLLECT MANUALLY applies to data acquisition requiring user actions, or that is intended to run for a long time. COLLECT USING SCRIPT applies to situations where you want to collect for a short time under script control. The second option provides a STOP button to terminate data acquisition.

Note that the Focus Error, Focus Sum, Focus Diff, Focus State, and Focus Control settings are considered as other head parameters. The term ‘focus information’ refers to data captured only when using the Focus Data setting.

### strokcal.spt

This script prompts you to choose one of two options. Option 1, Regular Strokecal, plots three stroke profiles—stroke init 33, stroke init 1, and the calibrated profile following stroke cal.

Option 1 is the preferred mode. Option 2, Stroke graph (without the stroke cal) plots three stroke profiles—stroke init 33, stroke init 1, and stroke init 16. Option 2 is useful for heads unable to perform a beam balance.

If you select Option 1 and beam balance fails, the script will automatically perform Option 2. The output plot shows any weak or stuck channels on the head and includes firmware version information and head parameters.

The script does a reset head at the end.

# Thermal 2 and Thermal 3 script details

**BBsprofile.spt**

**BBSprofile graphs**

**BBSProfileAll.ps** this graph (TH2) contains three elements:

1. Write laser profile with zero attenuation
2. Write laser profile with half (127) attenuation
3. Write laser profile with full (255) attenuation

**Ratio.ps** is a plot of elements c over a from **the BBSProfileAll.ps** graph.

**Sample graph**: Sample bbsprofile graph indicating a stuck pixel

**BBSProfile data**

There is one output file, **BBSProfile_data.txt**. This file contains five columns:

1. Pixel number
2. Write laser reading, zero attenuation
3. Write laser reading, half attenuation
4. Write laser reading, full attenuation
5. Ratio of full attenuation over zero attenuation

If the stroke width is less than 240, the write laser readings and the ratio for unused pixels are set to zero.

**Data interpretation**

A successfully balanced beam will have a flat profile on plate. In order to achieve this, the highest point of the full attenuation graph must be higher than the lowest point of the zero attenuation graph. You should have a large buffer between these two points, as this will allow for changes due to aging of the write laser and light valve. If the slope and curve are non-zero the final balanced profile will not be a straight horizontal line. This may end up using some of the buffer available for the aging of the laser.

**Note**: The final balanced profile does not need to be a straight horizontal line.

**BeamBalance.spt**

**BeamBalance graphs**

- **Sample graphs:** Good results | Poor results 1 | Poor results 2 | Poor results 3 | Failed example 1 | Failed example 2. If you get results that look like failed 1 or failed 2, escalate to the PSS.
- **BeamBalMaximum.ps**: shows the unattenuated write laser profile
- **BeamBalMinimum.ps**: shows the fully attenuated write laser profile
- **BeamBalMeasured.ps**: shows the write laser profile measured by the BBS when the beam is balanced (this balanced profile will not always be a flat line—it is the profile adjusted by the slope and curve settings)
- **Stroke.ps**: shows the stroke table used to obtain the results shown in BeamBalMeasured.ps
- **Ratio.ps**: shows the ratio of BeamBalMinimum.ps over BeamBalMaximum.ps
- **BeamBalResult.ps**: this graph shows a summary of the data obtained during the beam balance, consisting of:
    - BeamBalMaximum.ps
    - BeamBalMinimum.ps
    - BeamBalMeasured.ps
    - Stroke.ps

**BeamBalance data**

The one output file, **BeamBalance_data.txt**, contains seven columns:

1. Pixel number
2. Write laser profile, zero attenuation
3. Balanced write laser profile
4. Write laser reading, full attenuation
5. Target for the balanced write laser profile
6. Stroke table used to obtain the balanced beam
7. Ratio of full attenuation over zero attenuation.

If the stroke width is less than 240, no information is saved for the unused pixels.

**Data interpretation**

There should be a lot of buffer space between the minimum laser profile (full attenuation) and the balanced beam. This allows for aging of the laser and light valve.

A flatter stroke table indicates that the beam is fairly easy to balance. The Max Correction field at the bottom of each graph indicates the largest stroke value used.

The largest possible stroke setting is 255. If the maximum correction is close to this value, the write laser beam may become difficult to balance.

**BPMeasure.spt**

**BPMeasure graphs**

BPMeasure.ps: This graph shows the BP readings versus laser current.

**BPMeasure data**

There is one output file, BPMeasure_data.txt. This file contains nine data columns:

1. Laser current in Amps
2. Number of BP0 samples taken at this laser current, typically 128
3. The average BP0 sensor reading, in raw ADC units
4. The maximum BP0 sensor reading, in raw ADC units
5. The minimum BP0 sensor reading, in raw ADC units
6. Number of BP1 samples taken at this laser current, typically 128
7. The average BP1 sensor reading, in raw ADC units
8. The maximum BP1 sensor reading, in raw ADC units
9. The minimum BP1 sensor reading, in raw ADC units

**Data interpretation**

Check the graph for two things:

1. the sensor output should be reasonably linear
2. the sensor should not saturate

As the write laser does not start lasing right away (it tends to start around 10A), the graph will not run through the origin. The graph contains data for each BP sensor, BP0, and BP1.
If the BPD beam position setting is zero, these lines should overlap. If the reading at nominal laser current is less than 2000, then there could be problems with the sensor signal being too low.
Also, if the maximum reading is above 3500, then there could be problems with the sensor signal being too high. Ensure that the sensor isn't saturating.
**Note**: The M8 and BP readings are considered saturated if an increase in laser current does not produce an increase in the sensor reading. This tends to happen at a reading of 4095, but can occur at readings as low as or slightly lower than 3500.

**ContvBP.spt**

**ContvBP graphs** (all graphs have the Beam Position as the x-axis)

- ContOnOffvBP.ps: shows the contrast ratio in percent, the pixel-on power in Watts, and the pixel-off power in Watts.

- ContOnOffvBP_FIXED.ps:  this is the ContOnOffvBP.ps graph with the same vertical axis range as all the other FIXED.ps graphs.
- OffvBP.ps:  shows the write laser power with all pixels off, in Watts.
- OffvBP_FIXED.ps:  this is the OffvBP.ps graph with the same vertical axis range as all the other FIXED.ps graphs.
- OnvBP.ps:  shows the write laser power with all pixels on, in Watts.
- OnvBP_FIXED.ps:  this is the OnvBP.ps graph with the same vertical axis range as all the other FIXED.ps graphs.
- ContvBP.ps:  shows the contrast ratio, in percent.

TH2 Example:

- Example of TH2 good and poor results

TH3 Examples:

- ContvBP good
- ContvBP poor
- OffvBP good
- OffvBP poor

**ContvBP data** (there are two data files produced by this script)

ContvBP_data.txt: This file has four columns:

1. Beam position
2. Contrast ratio, in percent
3. Write laser power, in Watts, with all pixels on.
4. Write laser power, in Watts, with all pixels off.

Stroke_data.txt: This file contains the stroke table used when the data was taken.

**Data interpretation**

The peak of the on power tends to be about 1000 BP units less than that of the minimum contrast ratio. The BPD BeamPosition parameter should be set for peak power.

**ContvHV.spt**

**ContvHV graphs** (all graphs have the High Voltage as the x-axis)

ContOnOffvHV.ps:  shows the contrast ratio, in percent, the pixel on power, in Watts, and the pixel off power, in Watts.

ContOnOffvHV_FIXED.ps:  this is the ContOnOffvHV.ps graph with the same vertical axis range as all the other FIXED.ps graphs.

OffvHV.ps:  shows the write laser power with all pixels off, in Watts.

OffvHV_FIXED.ps:  this is the OffvHV.ps graph with the same vertical axis range as all the other FIXED.ps graphs.

OnvHV.ps:  shows the write laser power with all pixels on, in Watts.

OnvHV_FIXED.ps:  this is the OnvHV.ps graph with the same vertical axis range as all the other FIXED.ps graphs.

ContvHV.ps:  shows the contrast ratio, in percent.

**TH3 examples:**

- ContvHV
- OffvHV

**ContvHV data**

There is only one data files produced by this script,ContvHV_data.txt. This file has four columns:

1. High voltage
2. Contrast ratio, in percent

3. Write laser power, in Watts, with all pixels on.
4. Write laser power, in Watts, with all pixels off.

**Data interpretation**

The LVD HVTrim parameter should be set for minimum contrast ratio.

### FErrorvSR.spt

**Instructions**

This script assumes that a plate is on the drum in front of a properly installed head. You must turn on the focus servo as follows: at `lpos` = 0um (the most common location), enter the following commands: `th2> foc mode 1`, `th2>foc on`, `th2>foc lpos 0`. When the script ends, turn off the focus servo using the `foc off` command.

**FErrorvSR graphs**

FErrorvSR.ps: This graph shows the error, in raw units, versus surface reflectivity. The minimum and maximum error values are –32768 and +32767 respectively.

**Note**: To convert raw error units to microns, divide the raw number by the nvs setting `foc errorcal`.

FSumvSR.ps: This graph shows the sum versus surface reflectivity. The maximum sum reading is 2047.

**Sample graphs**: Good results | Bad results 1 | Bad results 2

**FErrorvSR data**

There is one output file, FErrorvSR_data.txt. This file contains three columns:

1. Surface reflectivity
2. Error, in raw units
3. Sum, in ADC counts (The sum is considered saturated if either an increase in the FOC RcvrGain parameter or a decrease in the surface reflectivity setting does not result in an increase in the sum signal. This tends to happen at 2047, but can happen with a sum signal several hundred lower.)

**Data interpretation**

The main purpose of this graph is to prove that the error reading does not change as the surface reflectivity setting is increased until the sum is saturated. If the error readings differ by more than a few microns while the sum signal is not saturated, then there are problems with the range finder. If this is the case, ensure that the FOC RcvrGain value is not too low.

### FocusActLin.spt

**FocusActLin graphs**

FocCurVLPosn.ps: This graph shows actuator travel, in raw units versus actuator current, in amps.
To convert raw lpos units to microns, divide the raw number by the nvs setting `foc lposcal`.

**FocusActLin data**

There is one output file, FocusActLin_data.txt. This file contains three columns: index, `lpos` in raw units, and current, in Amps.

**Data interpretation**

The range of `lpos` values should extend from between 1800 and 2000 at zero Amps, and below -1500 at 2A (except with version B board sets, where it may take almost 3A to travel below -1500).
The graph should be a backwards “S” shape, with little space between the two lines on the graph. The first line on the graph represents increasing current and the second line represents decreasing current.

### FocusPi.spt

**FocusPi graphs**

PI.ps - shows focus laser power versus focus laser current.

**Sample graphs**: Good results | Poor results

**FocusPi data**

There is one output file, FocusPI_data.txt. This file contains four columns:

1. Surface reflectivity

2. Focus laser power
3. Focus laser current
4. Linear regression residual power

**Data interpretation**

The PI graph should be a straight line, although it may deviate slightly from a straight line due to random measurement error. If the line is strongly curved or has a gross stairstep appearance, particularly at the high current end of the line, the focus laser may be damaged. The maximum focus laser current should not exceed 100 mA and rarely exceeds 60 mA. If the focus laser current exceeds 100mA at any point in the graph, the focus laser is damaged and the head will probably require replacement. Example of a normal laser. Example of a damaged laser

## LPosvLPos.spt

**LPosvLPos graphs**

LPosVLPos.ps: - shows actual lpos versus commanded lpos

**LPosvLPos data**

There is one output file, LPosVLPos_data.txt, containing three columns:
1. Commanded LPos, in microns.
2. Actual LPos, in microns.
3. This column contains no useful data.

**Data interpretation**

It is necessary for the focus servo to track the commanded input very accurately, to around 0.5um. If there is any ringing in the graph, or any audible ringing heard during the running of the script, then the servo parameters should be re-tuned using the ForcePing.spt script. The actual lens position will tend not to exceed roughly ±200um, since the Hall sensor will saturate at these extremities.

## BPDNoise.spt

More about BPDNoise.spt - When the BP servo is tuned using **BPDSettle.spt**, the BP sensor error signal must be stable to less than ±0.5%. If you are having trouble tuning the gains, it is possible that the sensor noise is too great.

**BPDNoise.spt** will test the BP sensor noise at various laser currents, from 15A up to nominal laser current, and display the results in the BPDNoise graph file.
The graphs axes are percent BP sensor error versus time. At each chosen write laser current, the script turns on the BP servo, lets the servo settle, and then waits for a few additional seconds.
The servo is then turned off at the time corresponding to 0's on the graph. Any noise at this point will be due to noise in the system, rather than servo setup problems.
The BP error signal will drift after the servo is turned off. This drift in the error can be either increasing or decreasing, but it must be smooth. Any noise in the system will be indicated by ripple superimposed on the drift.
The y axis will always have 1% divisions, so that it is easy to tell when the noise exceeds 0.5% (the point at which there will be problems successfully tuning the BP gains).

The raw data can be found in the **BPDNoise_data.txt** file. The organization of this file is discussed below.

- Ideally, the BPDNoise graph file will look like Example Graph 1. The slow drift of each line doesn't matter; however, there must not be any ripple on the graph.
- Example Graphs 2 and 3 show results with excess ripple. Example Graph 2 shows low frequency noise, while Example Graph 3 shows higher frequency noise. Neither is acceptable.
- Example Graph 4 shows noise only at lower currents. While this situation is not preferred, it may be acceptable if the given head tends to run only at higher write laser currents.
- Example Graph 5 shows marginal noise. In this condition, if **BPDSettle.spt** passes initially, there is no guarantee that things will remain acceptable.

If the output graph shows unacceptable noise and you are in production, call a production engineer for advice.

If you are at a customer site, and there are no plot artifacts, record that you made this observation, as this may make future debugging easier. If there are plot artifacts, call the Response Center for advice.

**Organization of BPDNoise_data.txt**

The first five elements of the first row of the data file contain the BPD NVS data: Kp, Ki, Kd, BeamPosition, and HoldCurrent, respectively.

The remaining rows contain the gathered data. The first column contains the time in seconds. The second through final columns contain the data for the increasing currents.

## BPDSettle.spt

The BPDSettle.spt script tests the BP servo performance for four current steps. If one of four tests produces a large settling time, you must tune the PID gains.
The goal of this operation is to have the percent error between the two BP sensor signals (this is the third graph on each of the BPDSettle.spt output windows) to settle within four seconds.

A settling time of 3 seconds is preferred, but this might be hard to achieve. The percent error is considered settled if the percent error is below 0.5%,
or the maximum difference between the two signals is two or less. The time at which the percent error is settled is called the settling time.
The BPDSettle.spt script will display a banner across the output window if you have not achieved these goals.

You might have to run BPDSettle.spt several times in order to find the correct gains. The following procedure is used to tune the PID gains:

1. **Adjust Kp**. If percent error graph takes too long to settle, increase Kp. It is not necessary to get the response completely settled in a maximum of 4 seconds, with 3 seconds preferred,
as this will be done in the next two steps. Aim for a settling time of roughly 8s to 10s - it is acceptable to have a lot of ringing in the response at the completion of this step.
Don't try too hard to achieve any settling time in the step, as the response is highly dependent on the value of Kd, which is adjusted in the next step. A typical plot at the end of this step can be found
in Example Graph 1. Try initial increments/decrements of 0.2 or 0.3. The final changes should only be 0.1.

2. **Adjust Kd**. Increasing Kd will provide a more damped percent error graph. Increase or decrease the value of Kd so that the actuator response is not ringing and settles in under 4s.
Try to have only one and a half or two periods of oscillation (for example, there are only 3 or four wiggles in the percent error graph). It is likely that it will be necessary to change the value of Kp
while tuning Kd in order to obtain this criteria. It is acceptable to have the response asymptotically decaying, as this will be fixed in the next step.

An example of an asymptotically decaying plot is shown in Example Graph 2. Note that there are only a couple of oscillations, but the percent error graph does not approach zero. While tuning Kd,
try initial increments/decrements of 0.2 or 0.3. The final changes should only be 0.1.

3. **Adjust Ki**. Increasing Ki will increase the asymptotic decay rate; however, increasing Ki too much will cause the response to oscillate. Change Ki until the settling time is less than 4s for all four write laser
current transitions. If you need to make minor adjustments to Kd or Kp in this step, that is acceptable. If you have to choose between two values of Ki, choose the lesser value. Try initial increments/decrements
of 0.01 or 0.02. The final changes should only be 0.005 or 0.01. An example of a good result can be found in Example Graph 3.

4. **Always run BPDSettle.spt as the final step**. Leave no set of final gains untested. Also, look at the three graphs for each of the four write laser transitions. Each subplot should be fairly smooth.
If one is excessively bumpy and you can't figure out how to fix the problem, contact a production engineer or the Response Center.

If you are unable to achieve acceptable gains, contact a production engineer or the Response Center.

### edgedetectfind.spt

The EdgeDetectFind.spt script:

1. Requires that a plate be loaded.
2. Asks for a best guess of edge strip position (`edge drumpos`).
3. Moves the carriage to the position before the plate edge.
4. Takes SUM measurements in a few sample drum positions.
5. Moves the carriage to the next step position and repeats the SUM measurements.
6. The SUM measurements are recorded with carriage and drum positions in a script result file, so they can be interpreted for the best edge strip position (`edge drumpos`).

### EdgeGroove3244.spt

EdgeGroove3244.spt - The problem in the 3244 application is that the edge detect groove is so close to the leading edge clamp that there is no way to get a leading-side signal from which to
derive the best edge detect drum position. By doing a second pass over plate you can observe where the leading edge limit is.

- The first pass must be done at a carriage position where there is no plate. The focus lens is contracted, for earliest exit of the focus beam from the edge detect groove.
- The second pass is done at a carriage position where there must be a plate. The focus lens is extended for latest clearing of the clamps. The script asks for the carriage positions.
  Note that you cannot do an edge detect, because it doesn't work yet.

To create the worst case conditions, the focus laser transmitter is turned on at full power and the focus receiver is at full gain for the edge detect groove pass.
Thus the focus SUM signal will rise to saturation as fast as possible. The on-plate pass is done at nominal settings for the active media.

The resulting graphs represent the limit possible for a given head install on a given machine. Select the operating drum position for edge detect in the middle of the O.K. area.

**Important**: The off-plate carriage position must be selected so that the focus beam end up over one of the wide grooves for the clamp bands, since the edge detect drum position must take into account
the case where the plate edge is over one of these grooves. The default off plate carriage position should put you over one of these grooves. This requirement reduces the margin for the drum position significantly.

In the example graphs, one was done properly, and one was done without the focus beam over a clamp band groove.

**Important**: There is somewhat of a problem in the case of the 3244 engine in that there is not a single EPOS that will satisfy all the possible conditions at once. See the third graph, where the on plate part of the graph was done at the operating point for the media (**Note**: The script has been modified to do this now also), preventing widening of the results due to finite focus beam width (it was also done at higher resolution).

Run this script any time a TH2 head is installed, or its mounting is adjusted, on a 3244 engine, or if you ever have troubles with image shifting in the subscan direction and towards the home position.

**Instructions**

1. Mount a plate.
2. Run the script.
3. Enter the best drum EPOS. For example:

```
>> set drum epos
NVS drum subsystem:
edge_position = 425  Saved = 425  Default = 450   tach ticks
Range: 0 to 36000
>> set drum epos 402
NVS drum subsystem:
edge_position = 402  Saved = 425  Default = 450   tach ticks
>> nvs save drum epos
Saving drum edge_position
```

## EdgeGrooveHydra.spt

EdgeGrooveHydra.spt - The problem in the 3244-type application is that the edge detect groove is so close to the leading edge clamp that there is no way to get a leading side signal from which to derive the best edge detect drum position. By doing the second pass over plate, you can observe where the leading edge limit is.

The first pass is done at a carriage position where there must be no plate. The focus lens is contracted, for earliest exit of the focus beam from the edge detect groove. The second pass is done at a carriage position where there must be a plate. The focus lens is extended for the latest clearing of the clamps. The script asks for the carriage positions. Note that you cannot do an edge detect, because it doesn't work yet.

To create the worst case conditions, turn the focus laser transmitter on at full power and the focus receiver at full gain for the edge detect groove pass. Thus the focus SUM signal will rise to saturation as fast as possible. The on-plate pass is done at nominal settings for the active media.

The resulting graphs represent the limit possible for a given head install on a given machine. Select the operating drum position for edge detect in the middle of the O.K. area.

**Important**: The off-plate carriage position must be selected so that the focus beam ends up NOT over one of the wide grooves for the clamp bands, since for Hydra the worst case seems to be not over the bands (this is different than for EdgeGroove3244.spt).

In the example graphs, one was done properly, and one was done with the focus beam over a clamp band groove.

Run this script any time a TH2 head is installed, or its mounting is adjusted, on a Hydra-type engine with an MCE host. Also, use it if you ever have troubles with image shifting in the subscan direction and towards the home position.

**Instructions**

1. Mount a plate.
2. Run the script.
3. Enter the best Edge DrumPos. For example:

```
NEWSENG>>
NEWSENG>> set edge
PRM edge parameters:
Group        Parameter       Active             Default            Units
-------------------------------------------------------------------------
edge         DrumPos         3.000              0.000              degrees
*** Command Complete
```

```
NEWSENG>>
NEWSENG>> set edge drumpos 3.2
'edge DrumPos' changed from 3.000 to 3.200 degrees
*** Command Complete
```

## EdgeGrooveQuantum800.spt

The EdgeGrooveQuantum800.spt script requires the following:

- There should not be a plate loaded. If a plate is loaded, it must not cover the drum at the carriage = 20 mm area.
- The host controller is assumed to be an MCE.

**Note**: The carriage position used has been selected to avoid confusion with the vacuum grooves in the same area.

**How this script works**

Two passes are run, one with the focus lens extended and the Z-stage forward and one with the focus lens contracted and the Z-stage back. This way it covers the two possible extremes for a given head installation.

To create the worst case conditions, the focus laser transmitter is turned on at full power and the focus receiver is at full gain. Thus the focus SUM signal will rise to saturation as fast as possible.

The resulting graphs represent the absolute limits possible for a given head install on a given machine. Select the operating drum position for edge detect in the middle of the O.K. area.

**Note**: There is an example output graph.

Run this script any time a TH2 head is installed or its mounting is adjusted, on a Quantum 800 engine, or if you ever have problems with image shifting in the subscan direction and towards the home position.

**Instructions**

1. Run the script.
2. Enter the best drum EPOS (the drum position midway between the no (or little) SUM signal region for the two cases). For example:

```
QUANTUM>> set edge
PRM edge parameters:
Group        Parameter       Active             Default            Units
-------------------------------------------------------------------------
edge         DrumPos         82.800             0.000              degrees
*** Command Complete
QUANTUM>> set edge drumpos 83.16
E PRM:     'edge DrumPos' changed from 82.800 to 83.160 degrees
'edge DrumPos' changed from 82.800 to 83.160 degrees
*** Command Complete
QUANTUM>> set edge
PRM edge parameters:
Group        Parameter       Active             Default            Units
-------------------------------------------------------------------------
edge         DrumPos         83.160             0.000              degrees
*** Command Complete
```

## EdgeMCE.spt

**When to run the script** EdgeMCE.spt

- If you are investigating focus errors at the plate edge.
- If you are having edge detect problems and image shift in the sub-scan direction towards the away end of plate and you want to know how the plate edge is.
- Anytime a sub-scan slice map of the plate (or drum) is desired.

**Instructions**

1. Have the desired plate for observation mounted and know your carriage positions for the start and end of mapping.
2. Decide on the desired resolution.
   **Note**: Typically, and for the case when you are going to map what the edge detect sequence would see, you should do an edge test first to set and hold the drum position.
   The script will ask if you want to move the drum or leave it where it is. However, you must then set up the focus reflectivity yourself because the head will have been set for the edge detect reflectivity which is different than for regular operation (use `th2 "foc reflect 0.12"`, for example). The script will do the rest.
3. Observe sample output graphs one and two for the results. Note the plate edge in the enclosed examples changes about 15 microns in 2 mm. This is very typical (changes of 40 microns in 2 mm have been observed) and by far exceeds the depth of focus of the write optics over one swath width. The SUM signal for the enclosed example is pretty good right to the plate edge. Sometimes the SUM signal rolls off at the plate edge, which in turn can cause the edge detect signal to delay triggering (example 3). The two examples are the same place on the same plate, but different sampling resolutions and overall coverage.

## EdgeMPE.spt

**When to run the script** EdgeMPE.spt

- If you are investigating focus errors at the edge.
- If you are having edge detect problems and image shift in the sub-scan direction towards the away end of plate and you want to know how is the plate edge.
- Anytime a sub-scan slice map of the plate (or drum) is desired.

**Instructions**

1. Have the desired plate for observation mounted and know your carriage positions for the start and end of mapping (in "CAU" Creo Arbitrary Units. CAU's must be used because it is the only way to get high-resolution sampling if required).
2. Decide on the desired resolution. **Note**: Typically, and when you are going to map what the edge detect sequence would see, you should first do a plot edge to set and hold the drum position.
   The script will ask if you want to move the drum or leave it where it is. However, you must then set up the focus reflectivity yourself because the head will have been set for the edge detect reflectivity which is different than for regular operation (use `th2 "foc reflect 0.13"`, for example (note some MPE versions do *not* adjust the SR for edge detection)). The script will do the rest.
3. Observe the output graphs for the results. Note the plate edge in the enclosed examples changes about 15 or 20 microns in one swath width. This is very typical (40 microns in one swath width have been observed) and far exceeds the depth of focus of the write optics over one swath width. The SUM signal for the enclosed example is pretty good right to the plate edge.
   Sometimes the SUM signal rolls off at the plate edge which in turn can cause the edge detect signal to delay triggering. The first example had the SUM signal too low. In the second, it is higher.

```
>> carriage
Querying Carriage Status ...
Current Position: 9791 mil (117500 counts) (valid)  <<<< from here 9791 / 117.5 = 83 Mils / CAU
Home Sensor:      False
Away Sensor:      False
Current Speed:    0 mil/sec (0 synth)
Carriage state: Idle
Most recent plate edge: 9430939 pmil                <<<< from here 9430.939 /83 = 113.6 CAU = edge
*** State message: Carriage command (8) Complete
```

## EndlessBBSProfile.spt

EndlessBBSProfile.spt continually performs a BBS PROFILE and displays the information in the Splot graph window. Simple statistics are also performed on the data.

The graph displays the first, most recent beam profile, maximum, minimum and average. The maximum, minimum and average are calculated for each pixel.

A dialog box is provided so that you can rescale the x-axis, issue a firmware command or quit. When quit is chosen, the most up to date graph and data are saved to the results folder.

Ideally, the EndlessBBSProfile graph file will look like Example Graph 1. The graphs should be reasonably consistent over time. If there is any drift, it should be small, and uniform across the swath.

This script is most useful to search for intermittently bad pixels. An example of an intermittently partially stuck off pixel is shown in Example Graph 2. The graph also shows a zoomed in view, so that it is clear which pixel is causing the problems.

**Organization of EndlessBBSProfile_data.txt**

The EndlessBBSProfile_data.txt file contains six columns:

1. Pixel number
2. Original profile
3. Most recent profile
4. Maximum sample
5. Minimum sample
6. Average

## flsrcheck.spt

If the script finds no compatibility problems you will see output similar to this:

Head Configuration Information:

```
==============================
CPLD Version = 7
T2HC Hdwr Version = 4
Focus Laser Driver = 2 'Toshiba'
Focus Laser = 2 'Toshiba'
Head firmware version = 1.04d
----> Focus laser and driver test : PASSED
----> Head firmware test : PASSED
------------------------------------------------------------------------
----> 'Overall test result' : PASSED
------------------------------------------------------------------------
FlsrCheck script finished.
```

If either of the two tests indicates FAILED, then the Overall test result will be FAILED and there is a compatibility problem.

If this happens to you, please check that you are using the version of TH2 head firmware in the latest controlled release.

## FocLog.spt

**When to run the script** FocLog.spt

Run this script whenever you want to extract and display head log data in the form that this script provides, which is simple sample verses, sample index around drum with each swath overlaid, and/or sample verses swath at one drum index. The plot also does contour plots. This script uses features only available in SPLOT version 3.95 and later. The script checks that your SPLOT version is recent enough (even if it is within Service Shell), and will abort if not.

**Instructions**

**Note**: If you want to log head data, you must already have the head log data captured before you run this script.

1. Use LogSetup.spt to setup the log for capture.
2. Plot or do plot headtest or whatever is needed to capture the log data.

3. Run this script and interact with the dialog boxes. It can take a long time over a dial-up connection to extract the entire log. Try to only ask for the amount of log needed for your application.
4. For your test, try to note roughly where around the drum your sample starts and ends and where the clamps are. This will help you to identify media verses bare drum verses clamp areas on the graphs.

Detailed instructions are available on TechPlanet.

## FocusvLPos.spt

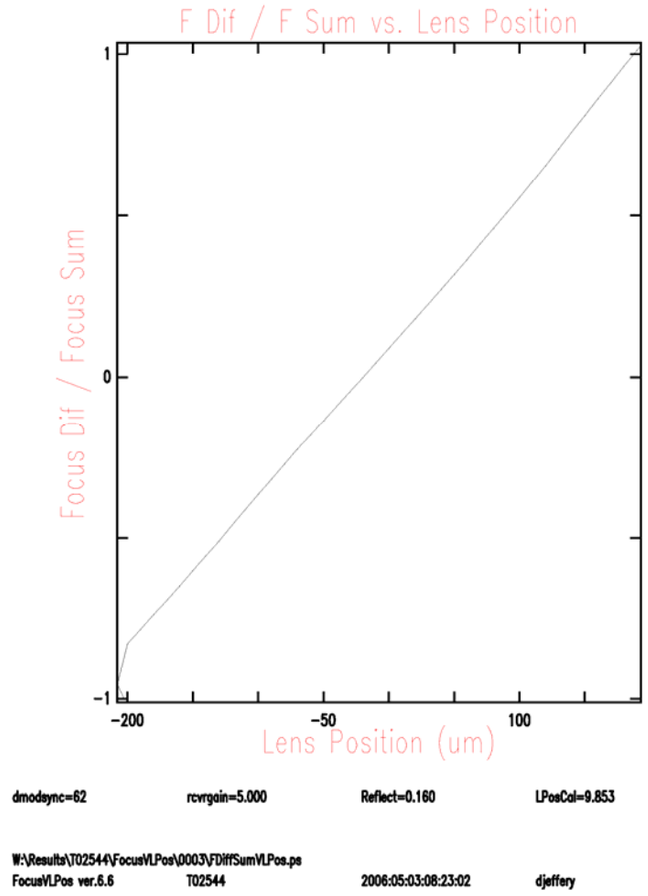
### Instructions

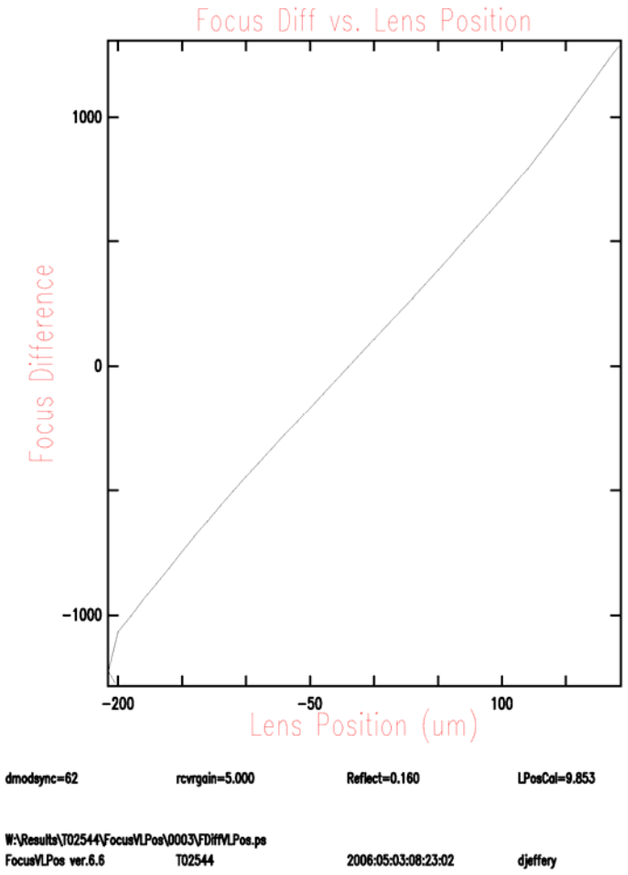
Note: For this script to work, a plate must be loaded and the head should be installed with an error of 0.
You must turn on the focus servo as follows: at `lpos` = 0um (the most common location), enter the following commands:
`th2> foc mode 1`, `th2>foc on`, `th2>foc lpos 0`. When the script ends, turn off the focus servo by typing `th2> foc off`.

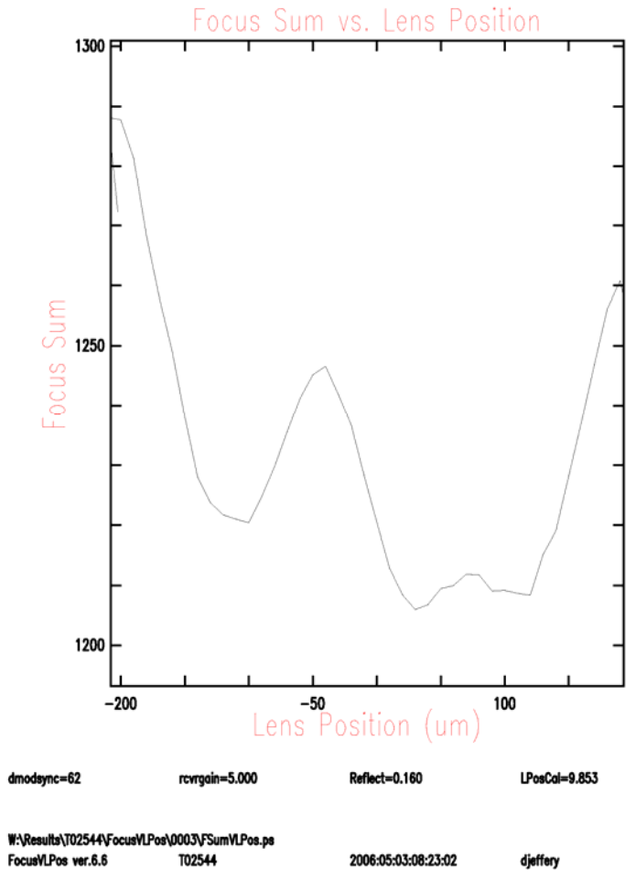
### FocusvLPos graphs

FDiffSumVLPos.ps: shows how the focus difference divided by the sum varies with lens position. . It should be a straight line with the line crossing the 0 value on both axis at the same time. The critical range is -150 to 150 microns (the range of the focus actuator).
FDiffVLPos.ps: shows how the focus difference varies with respect to lens position. It should be a straight line with the line crossing the 0 value on both axis at the same time. The critical range is -150 to 150 microns (the range of the focus actuator).
FSumVLPos.ps: shows how the different lens positions will affect the sum value seen by the focus system. Seeing some variation in this value is normal.

**Note**: These graphs were generated using firmware 1.01i.




**FocusvLPos data**

There is one output file, FocusVLPos_data.txt. The file contains eight columns:

1. Commanded lens position in microns.
2. The actual lens position in microns.
3. The focus difference in raw units.
4. The focus sum in raw units.
5. The focus difference divided by the focus sum.
6. This column contains no useful data.
7. This column contains no useful data.

8. This column contains no useful data.

**Data interpretation**

At least two things should be checked: first, the sum signal should not be saturated. The sum is considered saturated if either an increase in the `FOC RcvrGain` parameter or a decrease in the surface reflectivity setting does not result in an increase in the sum signal. This tends to happen at 2047, but can happen with a sum signal several hundred lower.

## FVdelay.spt

**Instructions**

This script assumes some type of media is on the drum in front of a properly installed head.
You must turn on the focus servo as follows: at `lpos` = 0um (the most common location), enter the following commands:
`th2> foc mode 1`, `th2>foc on`, `th2>foc lpos 0`. When the script ends, turn off the focus servo by typing `th2> foc off`.

**FVdelay graph**

A graph, FSumvDemodD.ps, shows the focus sum versus `dModSync`.

**FVdelay data**

The output file, FVDelay_data.txt, has three columns:

1. dModSync
2. Focus sum
3. Contains no useful data

The best `dModSync` value produces the highest focus sum.

## FVdelay_inverted.spt

**Instructions**

This script assumes some type of media is on the drum in front of a properly installed head.
You must turn on the focus servo as follows: at lpos = 0um (the most common location), enter the following commands:
th2> foc mode 1, th2>foc on, th2>foc lpos 0. When the script ends, turn off the focus servo using th2> foc off.

**FVdelay_inverted graph**

The graph, FSumvDemodD.ps, shows focus sum versus dModSync.

**FVdelay_inverted data**

The output file, FVDelay_inverted_data.txt, has three columns:

1. dModSync
2. Focus sum
3. Contains no useful data

The best dModSync value produces the highest focus sum.

## GraphPing.spt

The resulting lens position response is plotted, and saved as GraphPing.ps. Also, the bottom of the graph displays:

- start and end currents
- actuator parameters: the resonant frequency, the DC gain and the damping ratio
- controller gains
- estimator

The range for the autofocus parameters are:

- Resonant frequency: 120Hz to 170Hz
- DC Gain: 0.8 to 1.2
- Damping ratio: 0.01 to 0.08

Any parameter values outside these ranges might indicate a problem with the actuator. The step response is also saved in the text file, GraphPing_data.txt.

### HeadLogger.spt

The script performs the following actions:

1. Place a timestamp in the log file.
2. Set the laser to nominal current and record the nominal current in the log file.
3. Record the laser on time in the log file.
4. Set all lvd levels to zero and record the on power, off power and contrast ratio in the log file.
5. Balance the beam and record the on power, off power and contrast ratio in the log file.
6. Turn the laser off.

### LogSetup.spt

The default settings can be used to help determine focus-related problems, using foclog.spt. Each log item can be turned on or off by selecting the **Enabled** check box.

The divisor can be used to sub sample the data. That is, if you want only every third data item, then enter a divisor of three.

To exit the script, click **Exit with update** or **Exit without update**.

**Important**: Each log item takes a small amount of processing time in the head, so there is a maximum number of log items that you can have on at any one time. LogSetup.spt provides a **Check** button.

Once you have entered all of the desired log items, click the **Check** button. The script will do a quick calculation to see if you will be taking up too much processing time to save the log items.

If too many log items are chosen, then an error window will be displayed.

**Important**: It is best to turn off all log items once you have finished your task.

**Important**: Heads with firmware version 1.04 or later can handle Receiver Gain as well. LogSetup.spt will check for Receiver Gain and display as an option if it the firmware is a new enough version.

The updated dialog will look like this.

### LposStatsvLpos.spt

The LposStatsvLpos.spt script sweeps through all lens positions and samples a few hundred lens position sensor readings at each commanded lens position.

The script then displays a graph of the actual lens position versus commanded lens position as well as a graph of the standard deviation of the actual lens position versus the commanded lens position.

The standard deviation is a good measure of how well the lens is holding to the commanded lens position.

Ideally, the LposStatsvLpos graph file will look like Example Graph 1. The lower graph should not show any noise and the upper graph shouldn't exceed 0.5um noise, except at the extremes of travel.

If there is excessive noise, then the autofocus will have problems tracking the plate. An example of a bad result is shown in Example Graph 2. Note that the noise exceeds 0.5um in a significant part of the travel.

See also: good result | poor result

**Organization of LposStatsvLpos_data.txt**

The LposStatsvLpos_data.txt file contains six columns:

- The commanded lens position
- The instantaneous lens position
- The average lens position
- The standard deviation of the lens position
- The maximum lens position
- The minimum lens position

## M8Measure.spt

**M8Measure Graph:** M8Measure.ps shows the M8 readings versus laser current. See the following samples graphs: good result | poor result

**M8Measure Data:** The output file, M8Measure_data.txt, contains five data columns:

1. Laser current in Amps
2. Number of samples taken at this laser current, typically 128
3. The average M8 sensor reading, in raw ADC units
4. The maximum M8 sensor reading, in raw ADC units
5. The minimum M8 sensor reading, in raw ADC units

**Data interpretation:** Check the graph for two things: the sensor output should be reasonably linear, and the sensor should not saturate. As the write laser does not start lasing right away (it tends to start around 10A) the graph will not run through the origin. If the reading at nominal laser current is less than 2000, then there could be problems with the sensor signal being too low. Also, if the maximum reading is above 3500, then there could be problems with the sensor signal being too high. Ensure that the sensor isn't saturating. The M8 and BP readings are considered saturated if an increase in laser current does not produce an increase in the sensor reading. This tends to happen at a reading of 4095, but can occur at readings as low as or slightly lower than 3500.

## MPEOffsetPlot15.spt

**Before you run the script**

Before running the script you must manually select:

- media type
- main scan image size
- main scan shift.

Often, you will have to run a series with a coarse increment, followed by another series with a finer increment. This script uses a plot 12 (solid).

**When you run the script**

When you run the script, you will be asked to provide:

- initial carriage position (default 500mm)
- initial offset (default -120)
- final offset (default +120)
- offset increment (default 20).

Once these entries have been made, the script will plot the appropriate number of swaths.

## Parmcheck.spt

Run the Parmcheck.spt script whenever you want to check the head for any erroneous settings. The script also provides feedback on certain variables.

This script uses features only available in SPLOT version 3.95 and later. The script checks that your SPLOT version is new enough (even if it is within Service Shell), and will abort if not.

**Installation**

**Note**: For internal Creo Service Shell users, the script is already installed in the correct location.

The parmcheck.dll file must be in the proper folder (\ServiceShell\bin) for Service Shell or (\tii\splot\splot) for native field Splot.

The parmcheck.data file must be in the ...\th2\Includes folder.

**Note**: If you run the script, it will tell you where it looked for the file if it does not find it.

**Instructions**

Use NVSDump to obtain an up to date PRMParms.txt file.

Run the script and input the path to the PRMParms.txt file.

**Note**: Typically, this path would be the standard Thermal 2 path, but it is not a requirement.

Internal Example: W:\results\T00697\NVSDump\0004
Field Example: C:\results\T00697\NVSDump\0004
The script will automatically save an html and png files to the directory as the input path.

**HTML file of variables**

This script generates an html file to quickly and easily identify erroneous variables set on a Thermal 2 head. Some variables have help text to help solve problems.
The variables are flagged by 1 of 3 colors depicted their status.

- Green variables are acceptable.
- Orange variables are variables that should be looked at but may be okay.
- Red variables are errored and should be looked into closely.

There are a few different sections the variables fall into:

**Globally check variables**

These variables are shown in histograms of a large data set of other heads. The red bin is the one containing the value of the variable for the head checked in this script.
Hovering the mouse over the bins will show the number of heads in each bin. There are also some variables that will be flagged if they do not match a certain value that all the heads in the data set have.

**Single check variables**

These variables are checked against know valid values. Only variables that are flagged as errored or warnings are shown.

**LUT graphs**

These are graphs of LUT variables, the min and max acceptable ranges are shown on the graphs in red, and the actual line is shown in black.
The max and min are generated from the same data set as the histograms for the globally check variables.

## PingvCurr.spt

The actuator lens tube is mounted on a spring. When the actuator current is changed, the lens tube will move and oscillate about its final rest position.
These oscillations will decay until the lens tube comes to rest. This movement is shown in Example Graph 1.
The actuator movement can be described with three parameters:

- Resonant frequency - the frequency (how many cycles per second) of the oscillations.
- Damping ratio - a measure of how quickly the oscillations decay. The higher the damping ratio, the sooner the oscillations will decay.
- DC Gain - ratio of the change in position over the change in current. Due to the use of the autofocus LUT, the currents will not be in true Amps.

**PingvCurr graph**

Ideally, the PingvCurr graph file will look like Example Graph 2. The resonant frequency should start around 190-230Hz, depending on the dpi of the head, and will drop to around 120-160Hz around 1A, and then have another minor bump. The DC gain should be between 0.9 and 1.1. The damping ratio should be less than 0.08 for all currents. The damping ratio at lower currents will tend to be lower than at higher currents.

**PingvCurr data file**

The PingvCurr_data.txt file contains twelve columns:

1. The start current of the step
2. The final current of the step
3. The resonant frequency
4. The DC gain
5. The damping ratio
6. The controller gain K1 for a 100Hz closed loop bandwidth at this position

7. The controller gain K2 for a 100Hz closed loop bandwidth at this position
8. The controller gain Ki for a 100Hz closed loop bandwidth at this position
9. The estimator gain A1 for a 100Hz closed loop bandwidth at this position
10. The estimator gain A2 for a 100Hz closed loop bandwidth at this position
11. The estimator gain A3 for a 100Hz closed loop bandwidth at this position
12. The estimator gain A4 for a 100Hz closed loop bandwidth at this position

**ReadLogData.spt**

If extracting head data, it is assumed the head logs have been acquired and the data is in head memory.
This script is usually used where data is *not* drum position synchronous—FocLog.spt is usually used for drum position synchronous data.
The sample time is the number of samples divided by the total log acquisition time. This method of determining time access can be incorrect.
Samples of the dialog box: sample 1 | sample 2

**ReadPlotData.spt**

If extracting data from the head, then it is assumed that the head PLOT logs have already been acquired and the data is sitting in the head memory.
This script is used only for data from a plot log. The script is an interactive one, where you can plot graphs multiple times until you are done.
Use this script whenever you want to extract and display head log data from a plot, covering the whole plot and nothing outside of the plot.
Note that is sometimes preferred to use the mode of log acquisition that also samples the non-imaged area so that lead in and trail out effects can also be monitored.

**Instructions**

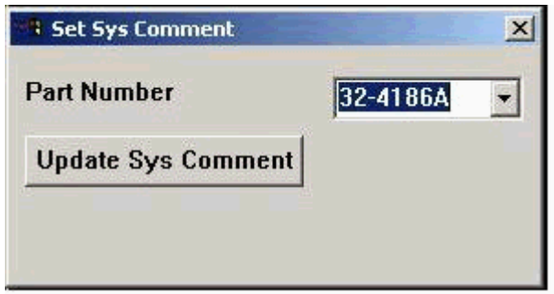
After a plot with the head in "LOG MODE PLOT" (see LogSetup.spt), run the script and interact with the dialog boxes.
**Note**: This script does not manage redirection properly. The head must already be in redirect mode.

**Data and graphs**

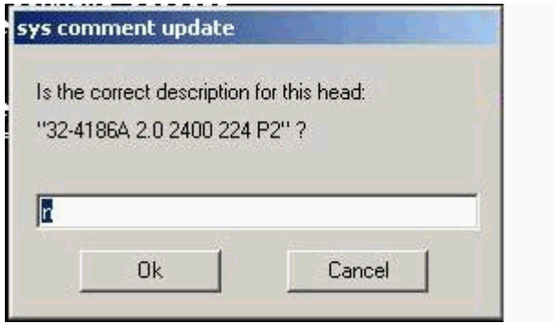
Example graphs and a dialog box.
In the first graph you can see the first swath has the focus beam partially off plate. This is normal.
In the second graph, it displays the first few swaths, in order to avoid clutter.
In this third graph, it does not look as if Focus is tracking as well as it should.
In this fourth graph, Focus seems to be tracking better.
The dialog box for the 4th graph.

**setsyscomment.spt**

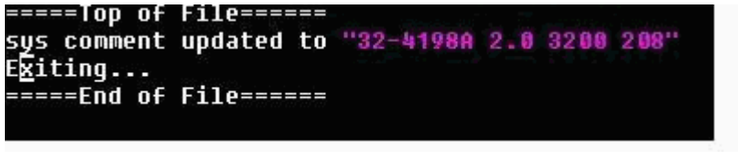
First you are prompted to choose the part number from a drop-down menu.




Next, you must confirm that the displayed description is valid. The default is n for no. You have to change it to y to confirm.




Upon successful completion, the output screen of your scripting application (Splot or Service Shell) displays a message acknowledging completion.




**Th2io.spt**

There are often bottlenecks in the system that reduce net I/O rate to unacceptable levels.

This script takes care of some head operations and rates the I/O in a consistent manner.

Use this script any time you want to check I/O performance. It is expected that this script will be more useful for system developers.

**Instructions**

1. Run the script.
2. A dialog box asks if you have a head data log already or if it should create a fake one.
   Do not choose to create a narrow printing log file unless you are specifically working on improving performance for short lines.
   **Note**: The I/O rate currently goes down by a factor of many times for short lines.
3. After the log has been acquired a dialog box asks what percentage of the log to extract, and what offset into the log to start at.
   **Tip:** Do not to ask for too much data if you are on a phone connection with limited bandwidth.

**Example script output**

```
ITH2IO: Test and rate head data output performance. Simple diagnostic tool.
TH2IO: Major Version: 1 Minor Version: 0
WARNING: 'Using test build firmware: Using f/w Version 9999.0' in d:\serviceshell\scripts\th2\includes\tii_special_io.spt, line 380.
TH2IO: log capture of fake data started...
TH2IO: log capture of fake data finished (or time out)...
TH2IO: Dumping data from head (might take a while)...
TH2IO: Head to workstation rate is: 4937 bytes per second.
TH2IO: Total Bytes (Net): 1135443 Time: 230 seconds.
TH2IO: Rating: Good.
```

## Th2ver.spt

**Instructions for** Th2ver.spt

Run the script. If the script crashes, you have a very old installation. Upgrade to the latest controlled release.

An example of the output:

```
Pass: Splot version: At least 3.950 needed. Version is 3.950
Pass: pm_include.spt version: At least 6.015 needed. Version is 6.015
Pass: stdtii_include.spt version: At least 6.800 needed. Version is 6.800
Pass: tii.spt version: At least 7.037 needed. Version is 7.037
Pass: tii_bbsprofile_include.spt version: At least 1.003 needed. Version is 1.003
Pass: tii_BG_include.spt version: At least 6.000 needed. Version is 6.000
Pass: tii_config_include.spt version: At least 8.016 needed. Version is 8.016
Pass: tii_config_include: TII_MINIMUM_SPLOT_VERSION should be at least 3.950. Now, it's: 3.950
Pass: tii_creep_include.spt version: At least 1.012 needed. Version is 1.012
Pass: tii_db.spt version: At least 2.007 needed. Version is 2.007
Pass: tii_foc_io.spt version: At least 6.206 needed. Version is 6.206
Pass: tii_foc_ping.spt version: At least 1.010 needed. Version is 1.010
Pass: tii_FocLog.spt version: At least 2.108 needed. Version is 2.108
Pass: tii_graph_include.spt version: At least 7.027 needed. Version is 7.027
Pass: tii_io_mpe.spt version: At least 6.000 needed. Version is 6.000
Pass: tii_io_primatives.spt version: At least 6.410 needed. Version is 6.410
Pass: tii_nvs_io.spt version: At least 6.311 needed. Version is 6.311
Pass: tii_port_include.spt version: At least 1.007 needed. Version is 1.007
Pass: tii_prm_include.spt version: At least 7.016 needed. Version is 7.016
Pass: tii_settingfile.spt version: At least 1.002 needed. Version is 1.002
Pass: tii_special_io.spt version: At least 6.725 needed. Version is 6.725

Info: LocalJig_include.spt: Serial port - use current settings on the remote machine
```

```
Info: LocalJig_Include.spt: This is a field configuration.
Info: LocalJig_include.spt: Save graph type is PNG (portable network graphics)
Info: LocalJig_include.spt: Use port switching for head redirection (at factory only)
```