GSM Async API 3.8.78.0

7/29/2024

A demo application of the GelSight Mobile Async API v3.8.78.0.

Objective

Describe the GelSight Mobile (GSM) Async API for external software developers.

Use Cases

The API is developed for two main use cases:

- Automation
- Extension of manual use of GSM

Automation

Enable the integration of GSM (GelSight Mobile) into a fully automated workflow where GSM and a GelSight sensor are controlled components in a robotic system. The intended scenario is in-line inspection in a manufacturing setup. Possible other components in the system could be a robot holding the GelSight sensor and maybe another robot that controls a conveyor belt where manufactured parts are inspected and then pushed forward.

From a system perspective, GSM and the sensor are considered components, and not the entity in charge of the manufacturing operation. The Async API is designed with that mindset.

Given that most operations in GSM are orders of magnitude longer than control loop operations like moving a robot and monitoring the move, the chosen service provider methodology is that of an event-based system, where the controlling entity subscribes to relevant events and subsequently orders GSM to start working on completing different tasks, where the success/failure is reported back as events. This methodology has the added benefit of allowing human interaction with the workflow, e.g. allowing a human to press the scan button and then use the event raised at completion of a scan to trigger the next operations managed by the controlling entity.

Extension of manual use of GSM

The GSM Async API will raise events to an external app each time an analysis method is completed and saved by GSM. This will be a rich event that contains all outputs and inputs for this analysis method. Reporting this data to an external app enables the application developer to develop custom logic that incorporates the results of multiple scans. It also allows the program logic to take input from other systems such as an ERP/SAP system or other customized workflow software used by the customer.

Some customers may have a desire to automatically forward results generated by GSM into their existing data systems. Prior to the API, this was only possible by manually transferring results from GSM into the customers own system. By writing a simple external app that integrates the GelSight Async DLL, the app will receive events with the results of any analysis. The developer can then easily decide which section of the outputs will be transferred to their own data system, either automatically or via customized UI.

Some customers need to trigger the scan based on their own workflow but perform the analysis inside GSM. This could be using a hardware foot pedal to trigger a scan or maybe some other software system that needs to trigger a scan. Using the API, a single line of code can be written to trigger a scan. It is even possible to instruct GSM what to name the scan and where to store it, e.g. on a shared network drive.

Environment

The GelSight Async API is delivered as a .NET DLL. The DLL can be linked into multiple programming languages as long as .NET Standard 2.0 or later is installed on the computer. The HelloWorld sample app includes WPF UI. For that to show .NET Framework 4.7.2 or later is needed.

The operating system supported is Microsoft Windows 11. Microsoft Windows 10 is supported until October 14, 2025.

The API requires a licensed GelSight Mobile to be installed.

The API communicates with GelSight Mobile via a TCP socket connection. This means the application developer has the choice of running the custom app with the DLL on the same computer as GSM or on a networked computer.

Getting Started

To enable API developers a quick and easy start, an open-source Visual Studio sample project has been developed with a “Hello World” application. The Hello World application is only about 300 lines of code that demonstrate all key functions of the API.

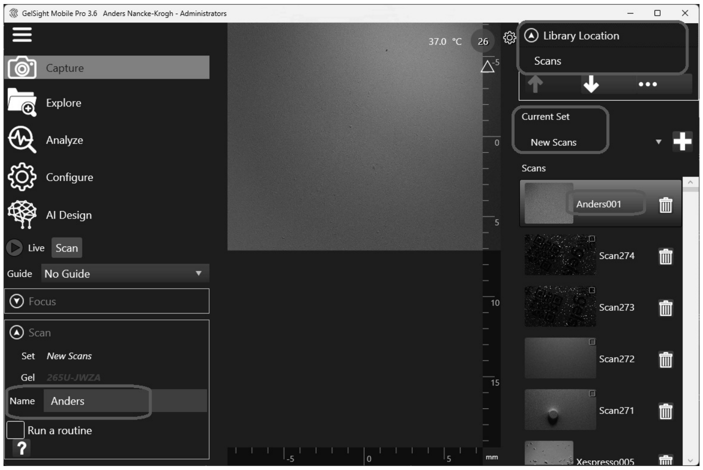
The sample is written in C#, but using simple structures so that non-C# developers can easily test the behavior and follow the logic.

1. Install Visual Studio 2022 Community and select ".NET desktop development" which enable WPF desktop apps.
2. Install and license GelSight Mobile 3.8 or later.
3. Enable the API Server in GSM Configure→System→API.
4. Download the GelSight Async API from the GelSight sdkdemo repo on GitHub.
5. Load and run the Visual studio project "GelSight.Api.HelloWorld.csproj".

Actions

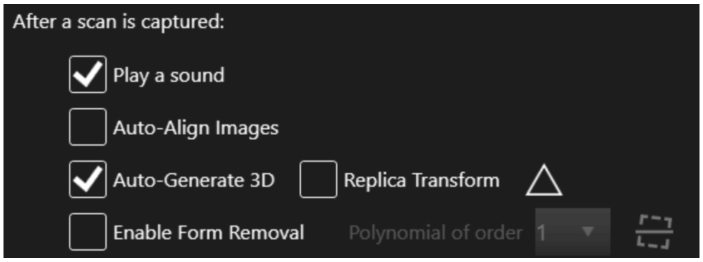
The following actions can be requested by the application developer via the DLL. Details of the parameters and data structures are documented in the sample code.

NOTE: Value types with a '?' are nullable in C#.

Name	Properties	Description
RequestScan	- ScanPath [string?] - ScanPrefixName [string?]	In GSM, the scan folder is created based on the following properties managed via the GSM UI. - Library Location (in this example "Scans") - Current Set (in this example "New Scans") - Name (in this example "Anders") The scan folder is created as a subfolder in <Library Location>. The scan folder is named <Name><enumerator>.  GelSight Mobile Pro 3.6 Anders Nancke-Krogh - Administrators Library Location Scans Current Set New Scans Set New Scans Gel 265U-JWZA Name Anders □ Run a routine Capture Explore Analyze Configure AI Design Live Scan Guide No Guide Focus Scan 37.0 °C 26 Scans Anders001 Scan274 Scan273 Scan272 Scan271 Xespresso005 -5 0 5 10 15 mm The parameters of this method are,

Name	Properties	Description
		• ScanPath, which is used to replace <Library Location>/<Current Set>. • ScanPrefixName, which is used to replace the name field in the lower left corner of the GSM user interface. The final name for the scan will be generated by GSM based on enumeration of the existing scan names in order to not create duplicate scan folders. If either the ScanPath or ScanPrefixName string is an empty string or null, then the scan will be saved using the current UI settings described above.
RequestAnalysis	• ScanFolder [string] • AnalysisName [string]	The ScanFolder parameter is the path to the scan data to be used by the analysis. The name of the analysis to run on the scan, is defined by the AnalysisName parameter. If the analysis name is incorrect or refers to an unlicensed analysis type, then the ErrorMessageReceived event will be raised by the ConnectionManager. This setup will support variants as well, so a GSM admin can create a variant of a baseline analysis method, modify the parameters, e.g. Pass/Fail to their needs and then make it available to the API. The client simply needs to pass the name of the variant analysis in the AnalysisName parameter. After running the analysis, the result is automatically saved and the ConnectionManager will raise the AnalysisSaved event. Critical note If the scan is already displayed in the “Analyze” view in GSM, any changes made by the user will be saved without notifying the user.

Name	Properties	Description
RequestLiveView	- Active [bool?]	This request will only change the live view status if a device is currently connected, and it is possible to navigate to the Capture view. Setting the Active parameter to false will turn the camera off to save power. Setting the Active parameter to true will navigate GSM to the Capture view and turn the camera on. Setting the Active parameter to null will have the same effect as setting it to true. **NOTE:** GSM can never guarantee a change to live view as multiple conditions could prohibit this from taking place, for example: - A scan is left open with changes in GSM. The user would need to save before a change can be made. - The sensor is disconnected. - The gel is not calibrated. - Other system error conditions like out of disk space, etc.
TryDeleteScan	- ScanFolder [string]	GSM will attempt to send the folder passed in by the ScanFolder parameter to the Windows Recycle bin. The ScanDeleted event will then be raised with the result.
RequestHeightmap	- ScanFolder [string] - Align [bool?] - Detrend [bool?] - Replica [bool?] - Detrend Order [int?] (1-4)	GSM refers to the 3D model of the surface as a “heightmap”. On disk this data is saved in a tmd file, which is a binary file containing the Z height value of each data point on the surface. GSM will raise the HeightmapStarted and the HeightmapCompleted events to notify the client the status of the height map creation. Setting the Align parameter to true will run the ‘Auto-Align Images’ algorithm when creating the

Name	Properties	Description
		heightmap. Setting the Align parameter to false will not run the ‘Auto-Align Images’ algorithm when creating the heightmap. Setting the Detrend parameter to true will run the ‘Form Removal’ algorithm when creating the heightmap. Setting the Detrend parameter to false will not run the ‘Form Removal’ algorithm when creating the heightmap. Setting the Replica parameter to true will run the ‘Replica Transform’ algorithm when creating the heightmap. Setting the Replica parameter to false will not run the ‘Replica Transform’ algorithm when creating the heightmap. Setting the Replica, Detrend, or Replica parameter to null will honor the GSM setting for this option as shown in the image below.  After a scan is captured: ☑ Play a sound ☐ Auto-Align Images ☑ Auto-Generate 3D ☐ Replica Transform ☐ Enable Form Removal Polynomial of order 1

Properties

The following properties are promised to be in sync with the GelSight Mobile software and the hardware.

NOTE: Value types with a ‘?’ are nullable in C#.

Name	Properties / Type	Description
Status	- IsSensorPresent [bool] - IsSensorLive [bool]	Readonly. The nullable properties (the ones with '?') will be null prior to a connection being established.

Name	Properties / Type	Description
	- IsScanPossible [bool] - Model [string?] - DisplaySerial [string?] - Serial [string?] - Configuration [string?] - FirmwareVersion [string?] - NumberOfChannels [int?] - ImageWidth [int?] - ImageHeight [int?]	Firmware: only present for S2 sensors. ImageWidth/ImageHeight: pixels
GelInfo	- Serial [string?] - Type [string?] - UseBy [DateTime?]	Readonly. Will contain null values until connected and data received from GSM
IsConnected	[bool]	Readonly
Port	[ushort]	Read: anytime Write: prior to attempting a connection
Host	[string]	Read: anytime Write: prior to attempting a connection
Password	[string]	Read: anytime Write: prior to attempting a connection

Events

The Client DLL’s ConnectionManager will raise the following events when state changes in GSM. For more information, please refer to the Hello World sample project on GitHub.

Name	Properties	Description
ConnectionStateChanged	- ConnectionState	This event is raised when the GSM connection state has changed. The

Name	Properties	Description
		possible connection states are: - Not Started - Connecting - Connected - Connected And Validated - Disposing - Validation Failed - Version Mismatch
StatusChanged	- IsSensorPresent [bool] - IsSensorLive [bool] - IsScanPossible [bool] - Model [string?] - Serial [string?] - Configuration [string?] FirmwareVersion [string?] - NumberOfChannels [int?] ImageWidth [int?] ImageHeight [int?]	This event is raised when, - A sensor is connected or disconnected. - Entering or leaving the GSM live view.. - The sensor ability to make a capture has changed.. Note that some of the properties can be null if their data is not defined. For example, the Serial parameter is an empty string when IsSensorPresent is false.
GelChanged	- Serial [string?] - Type [string?] - UseBy [DateTime?]	The event is raised when a gel is changed. Notice that only the Serial property is guaranteed to have a valid string.
ScanCompleted	- ScanFolder [string] - ScanMetaData [string]	The event is raised when a scan is completed and all files have been saved to the scan folder. The ScanFolder parameter is the folder where GSM has saved all files related to the scan. The ScanMetaData parameter is the content of the yaml file

Name	Properties	Description
		formatted as a JSON string. Refer to the ScanCompleted Event JSON section later in this document for more information.
HeightmapStarted	• ScanFolder [string]	The event is raised when 3D generation of scan data is started.
HeightmapCompleted	• ScanFolder [string] • Success [bool]	The event is raised when 3D generation of scan data is completed and has been saved to the file system.
AnalysisSaved	• ScanFolder [string] • Success [bool] Results [string] (json formatted) ErrorMsg [string] RequestId [int]	The event is raised when an analysis of a scan has been saved to the file system. The ScanFolder parameter is the folder where GSM saved all files related to the scan. The Success parameter indicates if the requested analysis was successful. The Results parameter is the contents of the scancontext.yaml file formatted as a JSON string. It contains the user created shapes as well as user inputs and generated outputs of all analysis methods run on this scan. The API caller will be able to retrieve PASS/FAIL information directly from the Results JSON string. The ErrorMsg parameter contains a string with information regarding

Name	Properties	Description
		why an analysis failed to run. The RequestId is the id of the client message used to initiate the analysis. It may be 0 if the analysis was not initiated remotely.
ScanDeleted	• ScanFolder [string] • Success [bool]	This event is raised after the TryDeleteScan request is made or if the user deletes a scan in GSM. The Success parameter represents if the deletion operation succeeded or failed. NOTE: If a scan is deleted outside of GSM, such as in Windows Explorer for example, the API will not raise this event.
ErrorMessageReceived	• ErrorMessage [string] • RequestId [int]	This event is raised if an error occurs in GSM while, • Capturing a scan. • Creating a heightmap. • Running an analysis method. • A hardware error has occurred. The RequestId is the id of the client action that was being processed. It may be 0 if the operation was not initiated by the Client program.

ScanCompleted Event

The ScanMetaData parameter in the ScanCompleted event contains JSON formatted data from the scan.yaml file located in the newly captured scan’s folder on disk. It is recommended to use the free C# Newtonsoft Json.NET library to help parse this information, but the API developer is free to use any method that they choose.

The JSON string does contain some replication of data.

Header	Example	Commen
version	2.0	Version of t json structu
guid	dd773e2f-e0f7-4dbc-94c9-350b7de4e28c	Unique Id
createdon	2023-07-19 10:42:21	Date/time o scan captur
mmperpixel	0.0069981396828	Resolution heightmap and images [mm per pixel]
sdkversion	3.8.0	Version of internal GS math librar used in analysis methods
crop	(0, 0, 0, 0)	If image alignment i enabled, th captured sc will be cropped to smaller size that fits a rectangle. T unit for cro [pixel]
scanwidth		Width of sc [pixel]
scanheight		Height of scan [pixels
images[]	• image01.png	Array with l of filename

Header	Example	Commen
	- image02.png - image03.png - image04.png - image05.png - image06.png	for 2D imag captured fo this scan. Most senso have 6 images.
activeheightmap	scan05.tmd	Filename fo the heightmap data file. Th heightmap an internall used 3D file It is structured similarly to the 2D ima files, but th pixel locatio does not represent a intensity as the 2D files instead the pixel value represents the Z heigh [mm] at the given XY location.
activenormalmap	scan05_nrm.png	Filename fo normal ma The normal map is an internally used file. It structured

Header	Example	Commen
		same way a the 2D ima files, but th pixel locatio does not represent a brightness level like in the 2D files instead the pixel value represents the surface normal at t given XY location
calib	C:\Users\Public\Documents\GelSight\Scans\scan05\Calib-2A3F-2JTC_20230606_1413.yaml	Path and filename fo calibration data
aligned	false	Alignment status (boo
cameratostage		Obsolete
calibration.date	2023-06-06 14:14:12	Date of calibration
calibration.username	TABLET-BTIUPLFS\GelSight	Windows username o the accoun used at tim of calibratio
camera.cameraid	CIMAF2239054	Camera (device) ser number
camera.cameratype	Ximea	Camera vendor

Header	Example	Commen
camera.gelid	2A3F-2JTC	Gel serial number
camera.lensfocuspos	-201.95	Focus settir NOTE: only present on sensors tha support autofocus
camera.shutter	0.689	Shutterspe [ms]
device.deviceconfigid	3	Unique id c the device model
device.devicefirmware	412	Firmware o the device NOTE: not present on sensor models
device.devicemodel	Series 2	Text description the sensor class
device.devicetemp	50.4	Camera temperatur [celcius]NC not present on all senso models
device.devicetype	0.5X 5 MP	Text description the sensor model (one sensor moc can have

Header	Example	Commen
		multiple device type
device.serialnumber	CIMAF2239054	Camera (device) ser number
metadata.appname	GelSight.Mobile	Sensor software interface te description
metadata.appversion	3.4.115.0	GSM versio used to capture the scan
metadata.gelid	2A3F-2JTC	Gel serial number
metadata.gelusecount	39	The numbe of scans taken with this gel
metadata.replica	false	Is replica enabled? [bool]
metadata.sdkversion	3.4.115.0	GSM versio used to capture the scan
metadata.username	TABLET-BTIUPLFS\GelSight	Windows username o the accoun used at tim of calibratio
metadata.detrended	false	Is form removal (detrending

Header	Example	Commen
		enabled? [bool]
metadata.detrendorder	2	The order c detrending (only valid i ‘detrended

AnalysisSaved Event

The Results parameter in the AnalysisSaved event contains JSON formatted data from the scancontext.yaml file located in the subfolder “analysis” inside the captured scan’s folder on disk. It is recommended to use the free C# Newtonsoft Json.NET library to help parse this information, but the API developer is free to use any method that they choose.

The JSON string contains two arrays.

- Shapes
- Routines

The “shapes” contain an entry for each user-created shape drawn on the scan.

The “routines” contain an entry for each analysis method that the user has saved for this scan.

Shapes

Shapes are defined in a coordinate system that has (0,0) located in the TopLeft corner of a scan.

Each shape has the following shared properties.

Header	Example	Comment
id	102750541	Unique id referring to this instance of shape [int]

Rectangle

Header	Example	Comment
type	Rectangle	Internal string reference to the type of shape [string]
name	Rectangle	User facing name of the shape [string]

Header	Example	Comment
x	1216.79012346	X position of TopLeft corner [pixel]
y	313.32345679	Yposition of TopLeft corner [pixel]
w	562.765432099	Width of rectangle [pixel]
h	1219.83209877	Height of rectangle [pixel]
rotation	0	Rotation [degree]

Ruler

Header	Example	Comment
type	Ruler	Internal string reference to the type of shape [string]
name	Ruler	User facing name of the shape [string]
x1	1216.79012346	X position of first point of ruler [pixel]
y1	313.32345679	Y position of first point ofruler [pixel]
x2	562.765432099	X position of second point ofruler [pixel]
y2	1219.83209877	Y position of second point ofruler [pixel]

Line

Header	Example	Comment
type	Line	Internal string reference to the type of shape [string]
name	Line	User facing name of the shape [string]
x1	1216.79012346	X position of first point of line [pixel]
y1	313.32345679	Y position of first point of line [pixel]
x2	562.765432099	X position of second point of line [pixel]
y2	1219.83209877	Y position of second point of line [pixel]

Circle

Header	Example	Comment
type	Circle	Internal string reference to the type of shape [string]
name	Circle	User facing name of the shape [string]
x	1216.79012346	X position of center [pixel]
y	313.32345679	Y position of center [pixel]
r	562.765432099	radius [pixel]

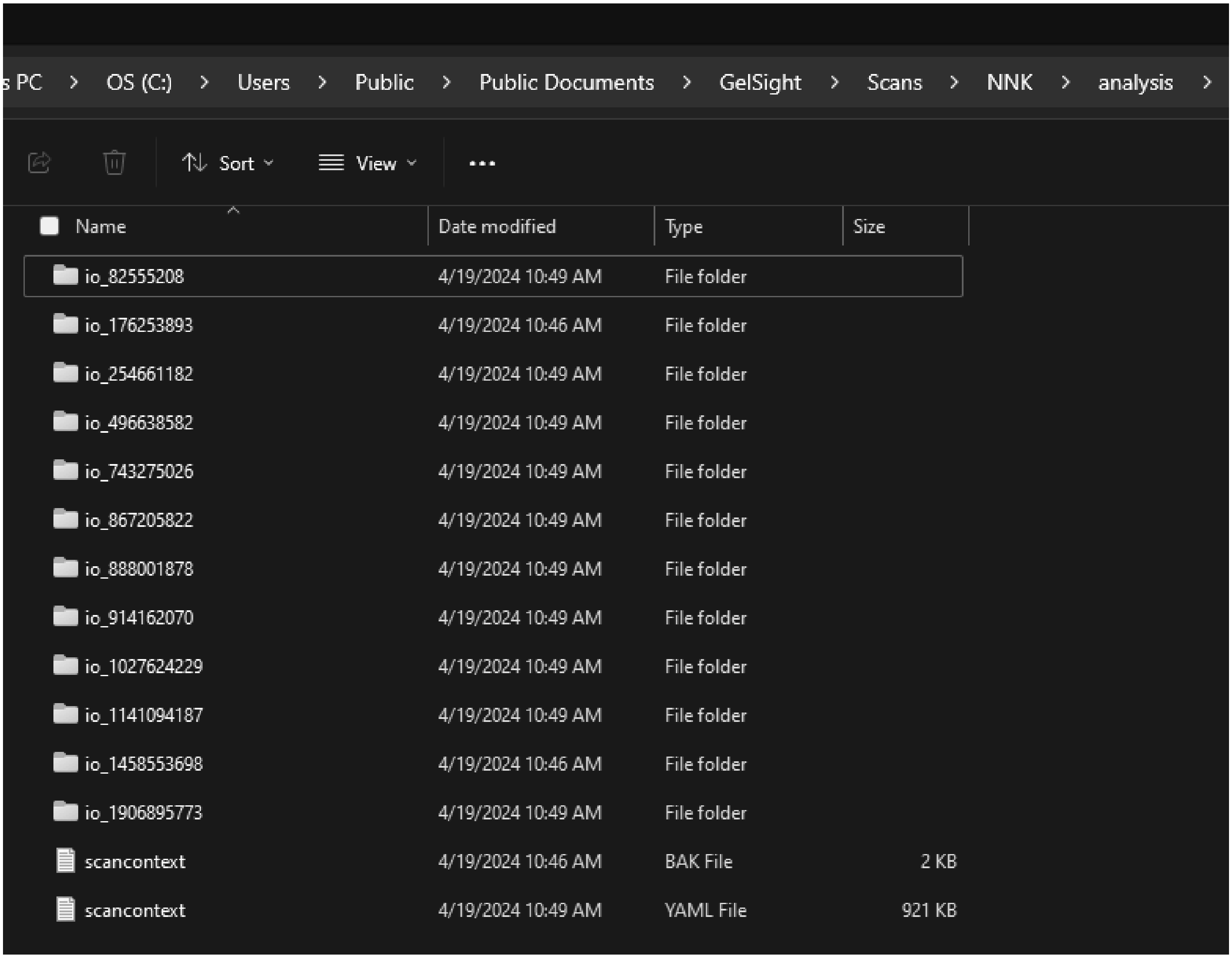
Polygon/Polyline

Header	Example	Comment
type	PolyLine	Internal string reference to the type of shape [string]
name	PolyLine	User facing name of the shape [string]
points[]	[(66.2933631391, 1300.69689919), (2348.81804603, 1367.46178936), (62.1205575031, 2001.72824603)]	Array of X/Y position of each point in the PolyLine [pixel]
closed	true	true: polygonfalse: polyline

Routines

Some routines will allow the user to draw one or more shapes and use them as input to the routine. If so, those inputs are referred to as shapeid which is referencing the array of “shapes”. One example is offset which requires a user drawn line as the shape input.

Some of the routines create a PNG output file that is rendered on top of the scan for visualization of the results. The filename is an output parameter from the routine, but to avoid overlap with other routines, each routine will get its own subfolder with the following naming convention: <scan_folder>/analysis/io_<routine_id>. Example is shown below for a scan named “NNK”.


s PC > OS (C:) > Users > Public > Public Documents > GelSight > Scans > NNK > analysis >
Sort
View
•••

Name	Date modified	Type	Size
io_82555208	4/19/2024 10:49 AM	File folder	
io_176253893	4/19/2024 10:46 AM	File folder	
io_254661182	4/19/2024 10:49 AM	File folder	
io_496638582	4/19/2024 10:49 AM	File folder	
io_743275026	4/19/2024 10:49 AM	File folder	
io_867205822	4/19/2024 10:49 AM	File folder	
io_888001878	4/19/2024 10:49 AM	File folder	
io_914162070	4/19/2024 10:49 AM	File folder	
io_1027624229	4/19/2024 10:49 AM	File folder	
io_1141094187	4/19/2024 10:49 AM	File folder	
io_1458553698	4/19/2024 10:46 AM	File folder	
io_1906895773	4/19/2024 10:49 AM	File folder	
scancontext	4/19/2024 10:46 AM	BAK File	2 KB
scancontext	4/19/2024 10:49 AM	YAML File	921 KB

ABS 1781

Header	Example	Comment
type	FastenerABS1781	Internal string reference to the type of routine [string]
id	1458553698	Unique id referring to this instance of routine
name	ABS 1781	User facing name of the routine [string]
annotationids		Reference to the shapes used in the routine. These may be local rulers.
diameter	8.75	Input: diameter [mm]
flushoffset	0.5	Input: Flushness offset [mm]

Header	Example	Comment
flushminlim	-0.08	Input: Flushness Min Limit [mm]
flushmaxlim	0.1	Input: Flushness Max Limit [mm]
headdishmin	0.158	Output: Head Dishing Min [mm]
perpendicularity	1.97412801048	Output: Perpendicularity [degree]
flushcircle	(1811.29474823, 753.969925059, 524.440141531)	Output: location of circle (X,Y,Radius) [pixel]
flushminpt	(1291, 826)	Output: Location of min point (X,Y) [pixel]. Rendered in UI as blue dot on scan overlay
flushmaxpt	(1291, 826)	Output: Location of max point (X,Y) [pixel]. Rendered in UI as red dot on scan overlay
image	image.png	Output: Filename for scan overlay for UI rendering
passfail	false	Output: was routine able to run, i.e. was enough inputs provided to allow a run [bool]
meta_passedanalysis	false	Output: result of pass/fail settings made by user [bool]
meta_failurereason	Pass value is `False` . Required value is `True`	Output: text description of pass/fail result [string]
meta_sdkversion	3.7.102.0	GSM version used to run this routine

ABS 2322

Header	Example	Comment
type	FastenerABS2322	Internal string reference to the type of routine [string]
id	176253893	Unique id referring to this instance of routine
name	ABS 2322	User facing name of the routine [string]

Header	Example	Comment
annotationids[]		Reference to the shapes used in the routine. These may be local rulers.
diameter	6.76	Input: diameter [mm]
corediameter	4.7	Input: core diameter [mm]
coreminlim	-0.1	Input: Break-Off Min Limit [mm]
coremaxlim	0.1	Input: Break-Off Max Limit [mm]
flushoffset	0.5	Input: Flushness offset [mm]
flushminlim	-0.08	Input: Flushness Min Limit [mm]
flushmaxlim	0.1	Input: Flushness Max Limit [mm]
flushmin	-0.0752567052841	Output: Flushness Min [mm]
flushmax	0.0198004096746	Output: Flushness Max [mm]
coremin	-0.654451012611	Output: Break-off Min [mm]
coremax	0.0789419040084	Output: Break-off Max [mm]
perpendicularity	0.852393517214	Output: Perpendicularity [degree]
flushminpt	(1663, 1067)	Output: Location of flush min point (X,Y) [pixel]. Rendered in UI as blue dot on scan overlay
flushmaxpt	(944, 511)	Output: Location of flush max point (X,Y) [pixel]. Rendered in UI as red dot on scan overlay
coreminpt	(1403, 682)	Output: Location of core min point (X,Y) [pixel]. Rendered in UI as blue dot on scan overlay
coremaxpt	(1131, 534)	Output: Location of core max point (X,Y) [pixel]. Rendered in UI as red dot on scan overlay
image	image.png	Output: Filename for scan overlay for UI rendering
meta_passedanalysis	true	Output: result of pass/fail settings made by user [bool]
meta_failurereason		Output: text description of pass/fail result [string]
meta_sdkversion	3.7.102.0	GSM version used to run this routine

Defect Detection

Header	Example	Comment
type	DefectDetection	Internal string reference to the type of routine [string]
id	1027624229	Unique id referring to this instance of routine
name	Defect Detection	User facing name of the routine [string]
annotationids[]		Reference to the shapes used in the routine. These may be local rulers.
scratchtype36	2	Input: Scratch type [enum]1=narrow2=wide3=large
autothreshold	true	Input: automatic set threshold [bool] (hidden in UI)
depththreshold	0.005	Input: depth threshold [mm] (hidden in UI)
levelwidth	0.35	Input: levelling width [mm] (hidden in UI)
leveldiscardwidth	0.2	Input: Level Discard Width [mm]
levelregions	'[(0, 0.35087546203, None), (1.26590362772, 1.60989917873, None)]'	Input: Array of leveling regions. Each entry has the following properties- Start (referring to position on profile line) [mm]- End (referring to position on profile line) [mm]- Type (None/Max/Min), will always be None for level regions
primaryshapeid	0	Input: Reference to the shapeID used as input. 0=no shape input
debug	false	Internal use
length	3.37853770121 mm	Output: Length of defect [string]
width	1.27010051581 mm	Output: width of defect [string]
area	1.10270439261 mm3	Output: area of defect [string]
depth	0.23071826702 mm	Output: depth of defect [string]
point	(1687.88679927, 768.239506643)	Output: Location of deepest point (x,y) [pixel]

Header	Example	Comment
image	image.png	Output: Filename for scan overlay for UI rendering
profile	'[(0, -0.0037239978619), …… 0.0182266798741)]'	Output: Array of points in the profile. Each point in the profile is organized as (X,Y) [mm]
minpt	(0.67423127998, -0.23071826702)	Output: The location and value of the lowest point in the profile (X,Y) [mm]
plot	'[(0.0206397330606, …… -0.109643088634)]'	Output: Array of points in the ‘plot’ profile. Each point in the profile is organized as (X,Y) [mm]
profilelines	'[(1300.97348203, …. 873.307388357)]'	Output: Array of 5 lines to be rendered on the scan overlay. They define the area of the defect and the profile line. Each line is defined by (x, y, width, length) [pixel]
meta_passedanalysis	true	Output: result of pass/fail settings made by user [bool]
meta_failurereason		Output: text description of pass/fail result [string]
meta_sdkversion	3.7.102.0	GSM version used to run this routine

Fastener

Header	Example	Comment
type	Fastener	Internal string reference to the type of routine [string]
id	1141094187	Unique id referring to this instance of routine
name	Fastener	User facing name of the routine [string]
annotationids[]		Reference to the shapes used in the routine. These may be local rulers.
diameter	8	Input: diameter [mm]

Header	Example	Comment
flushoffset	0.5	Input: Flushness offset [mm]
flushminlim	-0.08	Input: Flushness Min Limit [mm]
flushmaxlim	0.1	Input: Flushness Max Limit [mm]
refsurfdist	1	Input: [mm] Range: 0 to 5.0, Default: 1.0
refsurfwidth	3	Input: [mm] Range: 0.5 to 5.0, Default: 3.0
flushmin	-0.146005272865	Output: Flushness Min [mm]
flushmax	0.161588162184	Output: Flushness Max [mm]
headdishmin	-0.111065380275	Output: Head Dishing Min [mm]
perpendicularity	2.51877608606	Output: Perpendicularity [degree]
flushcircle	(1794.50996421, 751.757145099, 504.849922529)	Output: Location of flush circle (CenterX,CenterY, Radius) [pixel]. Rendered in UI
flushminpt	(1299, 853)	Output: Location of flush min point (X,Y) [pixel]. Rendered in UI as blue dot on scan overlay
flushmaxpt	(2253, 540)	Output: Location of flush max point (X,Y) [pixel]. Rendered in UI as red dot on scan overlay
image	image.png	Output: Filename for scan overlay for UI rendering
passfail	false	Output: was routine able to run, i.e. was enough inputs provided to allow a run [bool]
meta_passedanalysis	false	Output: result of pass/fail settings made by user [bool]
meta_failurereason	Pass value is `False` . Required value is `True` .	Output: text description of pass/fail result [string]
meta_sdkversion	3.7.102.0	GSM version used to run this routine

Fillet Relief

Header	Example	Comment
type	FilletRelief	Internal string reference to the type of routine [string]
id	82555208	Unique id referring to this instance of routine
name	Fillet Relief	User facing name of the routine [string]
annotationids	'[]'	Reference to the shapes used in the routine. These may be local rulers.
diameter	5	Input: diameter [mm]
mindeburr	0.2	Input: Min Deburr Width [mm]
minfillet	0.6	Input: Min Fillet Relief [mm]
minwidth	-0.46407	Output: Min Width [mm]
maxwidth	-0.31861	Output: Max Width [mm]
minheight	-0.30142	Output: Min Height [mm]
maxheight	-0.01676	Output: MaxHeight [mm]
holestate	Fillet Relief	Output: Hole State [string]
innercircle	(1437.23219534, 810.470397303, 369.359957758)	Output: Location of innercircle (CenterX,CenterY, Radius) [pixel]. Rendered in UI
minline	(1345.01283861, 447.818991122, 1361.7876679, 513.152637655)	Output: Location of min line (X1, Y1, X2, Y2) [pixel]. Rendered in UI
maxline	(1391.81149212, 1176.64030064, 1397.61578418, 1130.69459793)	Output: Location of max line (X1, Y1, X2, Y2) [pixel]. Rendered in UI
minpt	(1380.59772401, 508.94809197)	Output: Location of min point (X, Y) [pixel]. Rendered in UI
maxpt	(1126.70609124, 849.659906008)	Output: Location of max point (X, Y) [pixel]. Rendered in UI

Header	Example	Comment
image	image.png	Output: Filename for scan overlay for UI rendering
meta_passedanalysis	True	Output: result of pass/fail settings made by user [bool]
meta_sdkversion	3.7.102.0	GSM version used to run this routine

Hole Diameter

Header	Example	Comment
type	HoleByEdge	Internal string reference to the type of routine [string]
id	130343849	Unique id referring to this instance of routine
name	Hole Diameter	User facing name of the routine [string]
annotationids	'[]'	Reference to the shapes used in the routine. These may be local rulers.
edgesearch	0.6	Input: Edge Search Percentage
regionmode	top	Input: Region Model (top/bottom) [enum]
nominaldiameter	5.4	Input: Nominal Diameter [mm]
diameter	7.07919649057 mm	Output: Diameter [string]
circularity	0.109889499366 mm	Output: Circularity [string]
circle	(1065.74334886, 747.556295914, 514.483133639)	Output: Location of circle (CenterX, CenterY, Radius) [pixel]. Rendered in UI
minpt	(1560, 647)	Output: Location of min point (X, Y) [pixel]. Rendered in UI
maxpt	(1578, 839)	Output: Location of max point (X, Y) [pixel]. Rendered in UI

Header	Example	Comment
meta_passedanalysis	True	Output: result of pass/fail settings made by user [bool]
meta_sdkversion	3.7.102.0	GSM version used to run this routine

LGP EN6100

Header	Example	Comment
type	FastenerLGPEN6100	Internal string reference to the type of routine [string]
id	1906895773	Unique id referring to this instance of routine
name	LGP EN6100	User facing name of the routine [string]
annotationids	'[]'	Reference to the shapes used in the routine. These may be local rulers.
diameter	6.6	Input: diameter [mm]
flushoffset	0.5	Input: Flushness offset [mm]
flushminlim	-0.08	Input: Flushness Min Limit [mm]
flushmaxlim	0.1	Input: Flushness Max Limit [mm]
flushmin	-0.39892	Output: Flushenss Min [mm]
flushmax	0.08389	Output: Flushenss Max [mm]
headdishmin	-nan(ind)	Output: Head dishing Min [mm]
perpendicularity	4.981434	Output: Perpendicularity [degree]
flushcircle	(1253.21141275, 865.579579818, 402.257984959)	Output: Location of flush circle (CenterX,CenterY, Radius) [pixel]. Rendered in UI
flushminpt	(1641, 758)	Output: Location of flush min point (X,Y) [pixel]. Rendered in UI as blue dot on scan overlay
flushmaxpt	(867, 981)	Output: Location of flush max point (X,Y) [pixel]. Rendered in UI as red dot

Header	Example	Comment
		on scan overlay
image	image.png	Output: Filename for scan overlay for UI rendering
passfail	false	Output: was routine able to run, i.e. was enough inputs provided to allow a run [bool]
meta_passedanalysis	False	Output: result of pass/fail settings made by user [bool]
meta_failurereason	Pass value is `False` . Required value is `True` .	Output: text description of pass/fail result [string]
meta_sdkversion	3.7.102.0	GSM version used to run this routine

LGP EN6114

Header	Example	Comment
type	FastenerLGPEN6114	Internal string reference to the type of routine [string]
id	254661182	Unique id referring to this instance of routine
name	LGP EN6114	User facing name of the routine [string]
annotationids	'[]'	Reference to the shapes used in the routine. These may be local rulers.
diameter	8.75	Input: diameter [mm]
flushoffset	0.5	Input: Flushness offset [mm]
flushminlim	-0.08	Input: Flushness Min Limit [mm]
flushmaxlim	0.1	Input: Flushness Max Limit [mm]
flushmin	-0.0931385	Output: Flushenss Min [mm]
flushmax	0.15830711	Output: Flushenss Max [mm]
headdishmin	-0.1037149	Output: Head dishing Min [mm]

Header	Example	Comment
perpendicularity	1.97412801	Output: Perpendicularity [degree]
flushcircle	(1811.29474823, 753.969925059, 524.440141531)	Output: Location of flush circle (CenterX, CenterY, Radius) [pixel]. Rendered in UI
flushminpt	(1291, 826)	Output: Location of flush min point (X, Y) [pixel]. Rendered in UI as blue dot on scan overlay
flushmaxpt	(2279, 516)	Output: Location of flush max point (X, Y) [pixel]. Rendered in UI as red dot on scan overlay
image	image.png	Output: Filename for scan overlay for UI rendering
passfail	false	Output: was routine able to run, i.e. was enough inputs provided to allow a run [bool]
meta_passedanalysis	False	Output: result of pass/fail settings made by user [bool]
meta_failurereason	Pass value is `False` . Required value is `True` .	Output: text description of pass/fail result [string]
meta_sdkversion	3.7.102.0	GSM version used to run this routine

Offset

Header	Example	Comment
type	Offset	Internal string reference to the type of routine [string]
id	2116715901	Unique id referring to this instance of routine
name	Offset	User facing name of the routine [string]
annotationids	'[]'	Reference to the shapes used in the routine. These may be local rulers.

Header	Example	Comment
levelorder	0	Input: Form removal / levelling order [0,1,2,3,4]
levelregions	'[]'	Input: Array of leveling regions. Each entry has the following properties- Start (referring to position on profile line) [mm]- End (referring to position on profile line) [mm]- Type (None/Max/Min/Mean), will always be None for level regions
numprofiles	1	Input: Number of profiles [int]NOTE: this must be an odd number: 1/3/5/7 etc.
profilewidth	0.2	Input: Profile Width [mm]
reintegrate	false	Obsolete
primaryshapeid	1205877184	Input: Reference to the shape used in the routine. It must be a line
offsetregion1	(3.96097276059, 4.06253616471, Mean)	Input: Bottom region- Start (referring to position on profile line) [mm]- End (referring to position on profile line) [mm]- Type (Mean/Max/Min)
offsetregion2	(2.09811776797, 2.30124457621, Mean)	Input: Top region- Start (referring to position on profile line) [mm]- End (referring to position on profile line) [mm]- Type (Mean/Max/Min)
debug	false	internal
offset	0.40064612	Output: offset [mm]
width	1.81473697	Output: width [mm]
profile	'[(0, 0.026422787106), ... , -0.369323686395)]'	Output: Array of points in the profile. Each point in the profile is organized as (X,Y) [mm]
region1	'[(3.96630011343, -0.344152654986), ... , (4.06253616471, -0.369323686395)]'	Output: Array of points of the section of the profile that is covered by region1 (bottom). Each point in is organized as (X,Y) [mm]

Header	Example	Comment
region1pt	(4.01441813907, -0.356764410628)	Output: Location of bottom point (X, Y) [pixel]. Rendered in UI as blue dot on profile
point1	(1256.91977285, 849.440800471)	Output: Location of bottom point (X, Y) [pixel]. Rendered in UI as blue dot on scan overlay
region2	'[(2.10344512081, 0.0424179539838), ... , 0.0416897666568)]'	Output: Array of points of the section of the profile that is covered by region2 (top). Each point in is organized as (X,Y) [mm]
region2pt	(2.19968117209, 0.043881712457)	Output: Location of top point (X, Y) [pixel]. Rendered in UI as red dot on profile
point2	(994.096567835, 827.072868129)	Output: Location of top point (X, Y) [pixel]. Rendered in UI as blue red on scan overlay
profilelines	'[(675.522985995, 799.960222866, 1263.88858067, 850.033890498)]'	Output: Array of lines that can be rendered in the UI on top of the scan. Each line represent a profile line. The number of entries will be controlled by `numprofiles` .Each line defined as (x1, y1, x2, y2) [pixel]
meta_passedanalysis	True	Output: result of pass/fail settings made by user [bool]
meta_sdkversion	3.7.102.0	GSM version used to run this routine

Particle Detection

Header	Example	Comment
type	ParticleDetection	Internal string reference to the type of routine [string]
id	888001878	Unique id referring to this instance of routine
name	Particle Detection	User facing name of the routine [string]
annotationids	'[]'	Reference to the shapes used in the routine. These may be local rulers.
particledirection	Peaks	Input: Particle Type (Peaks/Pits) [enum]

Header	Example	Comment
featuresize	0.125	Input: Nominal Feature Size [mm]
clippingdistance	0.003	Input: Clipping Plane Distance [mm]
minamplitude	0.003	Input: Min Particle Amplitude [mm]
maxamplitude	2	Input: Max Particle Amplitude [mm]
minarticlediameter	0	Input: Min Particle Diameter [mm]
maxparticlediameter	10	Input: Max ParticleDiameter [mm]
primaryshapeid	0	Input: reference to shape set to limit ROI0=no shape input
particlecount	1136	Output: Number of Particles [int]
particlecountperarea	473749.62	Output: Number of Particles (Scaled) [int]
coverage	0.07565559	Output: Coverage [%]
analyzedarea	239.789110491 mm2	Output: Analyzed Area [mm2]
smallestdiameter	0.01097875	Output: Particle Minimum [mm]
largestdiameter	3.60360622	Output: Particle Maximum [mm]
maxparticleamplitude	0.33044797	Output: Max Particle Amplitude [mm]
cleanlinessscore	1001	Output: Cleanliness Score [int]
image	image.png	Output: Filename for scan overlay for UI rendering
maxpt	(1299, 344)	Output: Center Location of max point (X, Y) [pixels]
report		Output: The table header and all rows of the "report" tab in GSM is represented in a JSON data structure. The format is self explanatory if viewed in a debugger along with the GSM table output.
cc1246ereport		Output: The table header and all rows of the "CC1246E Report" tab in GSM is represented in a JSON data structure. The format is self explanatory if viewed in a debugger along with the GSM table output.

Header	Example	Comment
cc1246ehistogram		Output: The table header and all rows of the "CC1246E Histogram" tab in GSM is represented in a JSON data structure. The format is self explanatory if viewed in a debugger along with the GSM table output.
meta_passedanalysis	True	Output: result of pass/fail settings made by user [bool]
meta_sdkversion	3.7.102.0	GSM version used to run this routine

Pit Detection

Header	Example	Comment
type	PitDetection	Internal string reference to the type of routine [string]
id	496638582	Unique id referring to this instance of routine
name	Pit Detection	User facing name of the routine [string]
annotationids	'[]'	Reference to the shapes used in the routine. These may be local rulers.
featuresize	0.125	Input: Nominal Feature Size [mm]
depththreshold	0.003	Input: Clipping Plane Distance [mm]
mindepth	0.003	Input: Min Particle Amplitude [mm]
maxdepth	2	Input: Max Particle Amplitude [mm]
minpitdiameter	0	Input: Min Particle Diameter [mm]
maxpitdiameter	10	Input: Max ParticleDiameter [mm]
primaryshapeid	0	Input: reference to shape set to limit ROI0=no shape input
pitcount	2103	Output: Number of Particles [int]
coverage	0.05875127	Output: Coverage [%]
smallestdiameter	0.01097875	Output: Particle Minimum [mm]

Header	Example	Comment
largestdiameter	3.04545045	Output: Particle Maximum [mm]
maxpitdepth	0.51647782	Output: Max Particle Amplitude [mm]
nominaldiameter	0.125	Output: Nominal Feature Size [mm]NOTE: this is a replication of the input
image	image.png	Output: Filename for scan overlay for UI rendering
minpt	(1401, 678)	Output: Center Location of deepest pit (X, Y) [pixels]
report		Output: The table header and all rows of the “Report” tab in GSM is represented in a JSON data structure. The format is self explanatory if viewed in a debugger along with the GSM table output.
meta_passedanalysis	True	Output: result of pass/fail settings made by user [bool]
meta_sdkversion	3.7.102.0	GSM version used to run this routine

Profile Geometry

Header	Example	Comment
type	ProfileGeometry	Internal string reference to the type of routine [string]
id	1857757578	Unique id referring to this instance of routine
name	Profile Geometry	User facing name of the routine [string]
annotationids	'[]'	Reference to the shapes used in the routine. These may be local rulers.
levelorder	0	Input: Form removal / levelling order [0,1,2,3,4]
levelregions	'[]'	Input: Array of leveling regions. Each entry has the following properties: Start (referring to position on profile line) [mm] End (referring to position on profile line) [mm] Type (None/Max/Min/Mean), will always be None for level regions

Header	Example	Comment
region	(1.74037925428, 5.22113776283, None)	Input: region Start (referring to position on profile line) [mm] End (referring to position on profile line) [mm] Type (None)
primaryshapeid	1182189484	Input: Reference to the shape used in the routine. It must be a line
x1	1.74037925	Output: X1 [mm]
z1	0.04892514	Output: Z1 [mm]
x2	5.22113776	Output: X2 [mm]
z2	-0.4751283	Output: Z2 [mm]
distancex	3.48075851	Output: Distance X1-X2 [mm]
distancez	0.52405349	Output: Distance Z1-Z2 [mm]
distancexz	3.51998748	Output: Distance 1-2 [mm]
slope	0.15055727	Output: Slope (0-1) [%]
angle	8.56198947	Output: Angle [degrees]
lsslope	0.19144595	Output: LS Slope (0-1) [%]
lsangle	10.8379063	Output: LS Angle [degrees]
radius	3.71729561	Output: Radius [mm]
region1pt	(1.74037925428, 0.0489251445335)	Output: Location of left most point (X, Y) [pixel]. Rendered in UI as blue dot on profile
region2pt	(5.22113776283, -0.475128341615)	Output: Location of right most point (X, Y) [pixel]. Rendered in UI as red dot on profile
xdistanceline	(1.74037925428, -0.475128341615, 5.22113776283, -0.475128341615)	Output: Location of horizontal X distance line on profile (x1, y1, x2, y2) [mm]. Rendered in UI as line on profile
zdistanceline	(1.74037925428, 0.0489251445335,	Output: Location of vertical Z distance line on profile (x1, y1, x2, y2) [mm]]. Rendered in UI as line on profile

Header	Example	Comment
	1.74037925428, -0.475128341615)	
slopeline	(1.74037925428, 0.0489251445335, 5.22113776283, -0.475128341615)	Output: Location of diagonal line representing slope on profile (x1, y1, x2, y2) [mm]]. Rendered in UI as line on profile
lsslopeline	(1.74037925428, -0.0572237745789, 5.22113776283, -0.723600909066)	Output: Location of diagonal line representing least square slope on profile (x1, y1, x2, y2) [mm]]. Rendered in UI as line on profile
circle	(4.1251441316, 3.11708773989, 3.71729561006)	Output: Location of circle representing radius on profile (CenterX, CenterY, Radius) [mm]]. Rendered in UI as circle on profile
profile	'[(0, 0.0431844494982), (0.00687896938449, ..., (0.041273816307, 0.0405447581414)]	Output: Array of points of the section of the profile that is covered by the region. Each point in is organized as (X,Y) [mm]
meta_passedanalysis	True	Output: result of pass/fail settings made by user [bool]
meta_sdkversion	3.7.102.0	GSM version used to run this routine

Profile Roughness

Header	Example	Comment
type	Roughness	Internal string reference to the type of routine [string]
id	1474910097	Unique id referring to this instance of routine
name	Profile Roughness	User facing name of the routine [string]
annotationids	'[]'	Reference to the shapes used in the routine. These may be local rulers.
levelorder	0	Input: Form removal / levelling order [0,1,2,3,4]

Header	Example	Comment
levelregions	'[]'	Input: Array of leveling regions. Each entry has the following properties: Start (referring to position on profile line) [mm] End (referring to position on profile line) [mm] Type (None/Max/Min/Mean), will always be None for level regions
reintegrate	false	Obsolete
primaryshapeid	213149554	Input: Reference to the shape used in the routine. It must be a line
lambdac	0.8	Input: Filter Cutoff lambda-c [mm]
numprofiles	1	Input: Number of profiles [int]NOTE: this must be an odd number: 1/3/5/7 etc.
profilewidth	1	Input: Profile Width [mm]
scale	1	Input: scale [double]
averageravalue	8.2036062	Output: Average Ra [um]
Ra	8.2036062	Output: Ra [um]
Rq	11.502679	Output: Rq [um]
Rp	28.2195084	Output: Rp [um]
Rv	37.1566086	Output: Rv [um]
Rz	30.5532495	Output: Rz [um]
Rt	65.376117	Output: Rt [um]
Rsk	-0.3781205	Output: Rsk [um]
Rku	4.59679143	Output: Rku [um]
RSm	791.520557	Output: Rsm [um]
Rc	24.5678074	Output: Rc [um]
evaluationlength	3.63826281	Output: Evaluation Length [mm]
scalefactor	1	Output: Scale [double]. This is a replication of the input

Header	Example	Comment
rprofile	'[(0.405779784215, 0.00052973614822), ..., (4.04404259523, 0.0202368823169)]'	Output: Array of points along the profile that represent the Roughness Profile. Each point in is organized as (X, Y) [mm]
wprofile	'[(0.405779784215, 0.0806891284185), ... , (4.04404259523, 0.162517905647)]'	Output: Array of points along the profile that represent the Waviness Profile. Each point in is organized as (X, Y) [mm]
pprofile	'[(0, 0.0138198800394), ... , (4.44982237944, 0.279243363504)]'	Output: Array of points along the profile that represent the Primary Profile. Each point in is organized as (X, Y) [mm]
profilelines	'[(984.310603058, 779.096194686, 1631.09547664, 779.096194686)]'	Output: The line defining the trace-line, will be the same as the input shape coordinates (x1, y1, x2, y2) [pixels]
meta_passedanalysis	True	Output: result of pass/fail settings made by user [bool]
meta_sdkversion	3.7.102.0	GSM version used to run this routine

Radius

Header	Example	Comment
type	Radius	Internal string reference to the type of routine [string]
id	1912752174	Unique id referring to this instance of routine
name	Radius	User facing name of the routine [string]
annotationids	'[]'	Reference to the shapes used in the routine. These may be local rulers.
levelorder	0	internal
levelregions	'[]'	internal

Header	Example	Comment
numprofiles	1	Input: Number of profiles [int]NOTE: this must be an odd number: 1/3/5/7 etc.
profilewidth	0.2	Input: Profile Width [mm]
linearthreshold	0	internal
reintegrate	false	Obsolete
primaryshapeid	491633211	Input: Profile Width [mm]
region	(0.86686880433, 2.60060641299, None)	Input: region on profile Start (referring to position on profile line) [mm] End (referring to position on profile line) [mm] Type (None)
radius	2.83856157	Output: Radius [mm]
meanradius	2.83856157	Output: Mean Radius [mm]
minradius	2.83856157	Output: Min Radius [mm]
maxradius	2.83856157	Output: Max Radius [mm]
medianradius	2.83856157	Output: Median Radius [mm]
centerprofile	'[(0, 0.0653711264391), ... , (3.45458003393, -0.580902561051)]'	Output: Array of points along the profile that represent the section of the profile covered by the region. Each point in is organized as (X, Y) [mm]
profiles	[]	Output: Profile List List of profiles produced by the routine not including the center profile. Each profile consists of points. Each Point is the T value (distance in mm from start of line) and the Z value (distance in MM from 0)
centercircle	(0.52594731328, -2.71080230828, 2.83856157236)	Output: The radius on the profile represented as a circle. (CenterX, CenterY, Radius) [mm]
circles	'[]'	Output: Circles The circles that fit each profile in ‘profiles’. Each circle consists of a

Header	Example	Comment
		center X, center Y and radius in image pixels.
profilelines	'[(963.446574878, 637.220803062, 1447.49202865, 770.750583414)]'	Output: Array of lines that can be rendered in the UI on top of the scan. Each line represents a profile line. The number of entries will be controlled by `numprofiles` . Each line defined as (x1, y1, x2, y2) [pixel]
meta_passedanalysis	True	Output: result of pass/fail settings made by user [bool]
meta_sdkversion	3.7.102.0	GSM version used to run this routine

Scratch

Header	Example	Comment
type	Scratch	Internal string reference to the type of routine [string]
id	785976931	Unique id referring to this instance of routine
name	Scratch	User facing name of the routine [string]
annotationids	'[]'	Reference to the shapes used in the routine. These may be local rulers.
tolerance	0.005	Input: Tolerance [mm]
filtersg	0.5	Input: Filter Sigma [mm] Default value: 0.5
polyorder	4	Input: Polynomial order [1-4]
shapeids	[1156429806, 1114464821]	Input: Reference to the two shapes used in the routine. All shapes must be of a closed type, i.e. no lines and no open polylines
debug	false	internal
minz	-0.6927297	Output: Min [mm]
maxz	0.07521429	Output: Max [mm]
minpt	(1403, 682)	Output: Min point location (X, Y) [pixel])

Header	Example	Comment
maxpt	(1116, 529)	Output: Max point location (X, Y) [pixel]
meta_passedanalysis	True	Output: result of pass/fail settings made by user [bool]
meta_sdkversion	3.7.102.0	GSM version used to run this routine

Shotpeen

Header	Example	Comment
type	ShotPeen	Internal string reference to the type of routine [string]
id	743275026	Unique id referring to this instance of routine
name	Shot Peen	User facing name of the routine [string]
annotationids	'[]'	Reference to the shapes used in the routine. These may be local rulers.
diameter	0.3	Input: Dent Diameter [mm]
depth	-0.001	Input: Min Depth [mm]
expand	0.35	Input: Dent Expansion [mm]
primaryshapeid	0	Input: Reference to the shapeID used as input. 0=no shape input
coverage	0.8733317	Output: Coverage [%]
image	image.png	Output: Filename for scan overlay for UI rendering
meta_passedanalysis	True	Output: result of pass/fail settings made by user [bool]
meta_sdkversion	3.7.102.0	GSM version used to run this routine

Surface Roughness

Header	Example	Comment
type	SurfaceRoughness	Internal string reference to the type of routine [string]
id	867205822	Unique id referring to this instance of routine

Header	Example	Comment
name	Surface Roughness	User facing name of the routine [string]
annotationids	'[]'	Reference to the shapes used in the routine. These may be local rulers.
primaryshapeid	0	Input: Reference to the shape used in the routine to define ROI. It must be a closed shape0=no shape used
secondaryshapeid	0	Input: Reference to the shape used in the routine to define an exclusion area within the ROI. It must be a closed shape0=no shape used
scale	1	Input: scale [double]
formremoval	None	Input: Form Removal (None, Poly1, Poly2, Poly3, Poly4) [enum]
lambdac	0.8	Input: Filter Cutoff lambda-c [mm]
applyfilter	true	Input: Apply Filter [bool]
Sa	2.18147096	Output: Sa [um]
Sq	4.61606018	Output: Sq [um]
Sp	58.5793331	Output: Sp [um]
Sv	96.4832976	Output: Sv [um]
Ssk	-2.3873798	Output: Ssk [um]
Sku	36.155147	Output: Sku [um]
Sz	155.062631	Output: Sz [um]
mask	mask.png	Output: Filename for scan overlay for UI rendering
scalefactor	1	Output: Scale [double]. This is a replication of the input
meta_passedanalysis	True	Output: result of pass/fail settings made by user [bool]
meta_sdkversion	3.7.102.0	GSM version used to run this routine

Weld Bead

Header	Example	Comment
type	Weld	Internal string reference to the type of routine [string]
id	914162070	Unique id referring to this instance of routine
name	Weld Bead	User facing name of the routine [string]
annotationids	'[]'	Reference to the shapes used in the routine. These may be local rulers.
estwidth	3.72	Input: Dent Diameter [mm]
diameter	135	Input: Min Depth [mm]
margin	0.7	Input: Dent Expansion [mm]
length	5	Input: Reference to the shapeID used as input. 0=no shape input
referencewidth	2.5	Output: Coverage [%]
referenceshape	Plate	Output: Filename for scan overlay for UI rendering
height	0.816874089636 mm	Output: Height [string]
width	3.06976611597 mm	Output: Width [string]
offset	0.565113858693 mm	Output: Planar Mismatch [string]
depth	0.0265661236748 mm	Output: Undercut Depth [string]
profile1	'[(0, -0.125322695717),,,... (0.041306122449, -0.113199078161)]	Output: Array of points defining Profile 1. Each point in is organized as (X, Y) [mm]
profile2	'[(0, -0.266413164522), ... (0.041306122449, -0.259776774578)]	Output: Array of points defining Profile 2. Each point in is organized as (X, Y) [mm]

Header	Example	Comment
mask	mask.png	Output: Filename for scan overlay for UI rendering
leftpoint1	(2.54721088435, -0.112809776873)	Output: Location of left point on Profile 1 (X, Y) [mm]
rightpoint1	(6.31983673469, 1.05287165696)	Output: Location of right point on Profile 1 (X, Y) [mm]
leftpoint2	(2.54721088435, -0.599049738613)	Output: Location of left point on Profile 2 (X, Y) [mm]
rightpoint2	(6.31983673469, 0.0660026953761)	Output: Location of right point on Profile 2 (X, Y) [mm]
centerpoint1	(10.12, 1.49889042827)	Output: Location of center point on Profile 1 (X, Y) [mm]
centerpoint2	(5.64517006803, 0.121794203458)	Output: Location of center point on Profile 2 (X, Y) [mm]
meta_passedanalysis	True	Output: result of pass/fail settings made by user [bool]
meta_sdkversion	3.7.102.0	GSM version used to run this routine

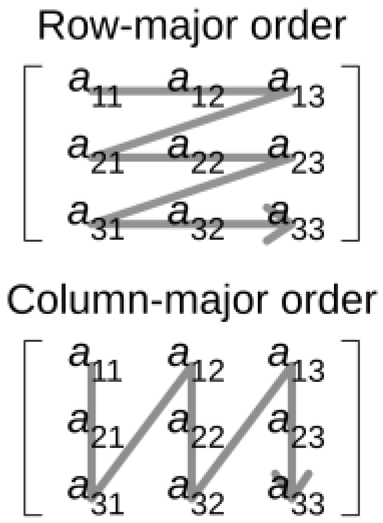
Heightmap Data File Format (.tmd)

Field	Name	Data Type	Size	Example	Comments
1	Signature	char[32]	32	“Binary TrueMap Data File v2.0\r\n\0”	Needs to be “Binary TrueMap Data File v2.0\r\n\0”.
2	Comment	char[]	Unknown	“The comment field goes here.”\0	Starting at the 33 byte in the file, read until the next null char (‘\0’). This field is at least one byte in length because it contains a null char.

Field	Name	Data Type	Size	Example	Comments
3	Width	int	4	256	Number of columns in the heightmap (x-axis).
4	Height	int	4	256	Number of rows in the heightmap (y-axis).
5	X Length	float	4	1.5	Length along the x-axis in mm.
6	Y Length	float	4	1.2	Length along the y-axis in mm.
7	X Offset	float	4	0.0	X-axis minimum value in mm.
8	Y Offset	float	4	0.0	Y-axis minimum value in mm.
9	Z height matrix	float[Width * Height]	4 * Width * Height		The Z height values in mm representing the surface in row-major order.

Notes:

- int types are signed 4-byte values.
- float is a 4-byte floating point number.
- A value of -1e10 represents a non-measured point on the surface. (Not used by GelSight Mobile).


Row-major order
a_{11} a_{12} a_{13}
a_{21} a_{22} a_{23}
a_{31} a_{32} a_{33}
Column-major order
a_{11} a_{12} a_{13}
a_{21} a_{22} a_{23}
a_{31} a_{32} a_{33}