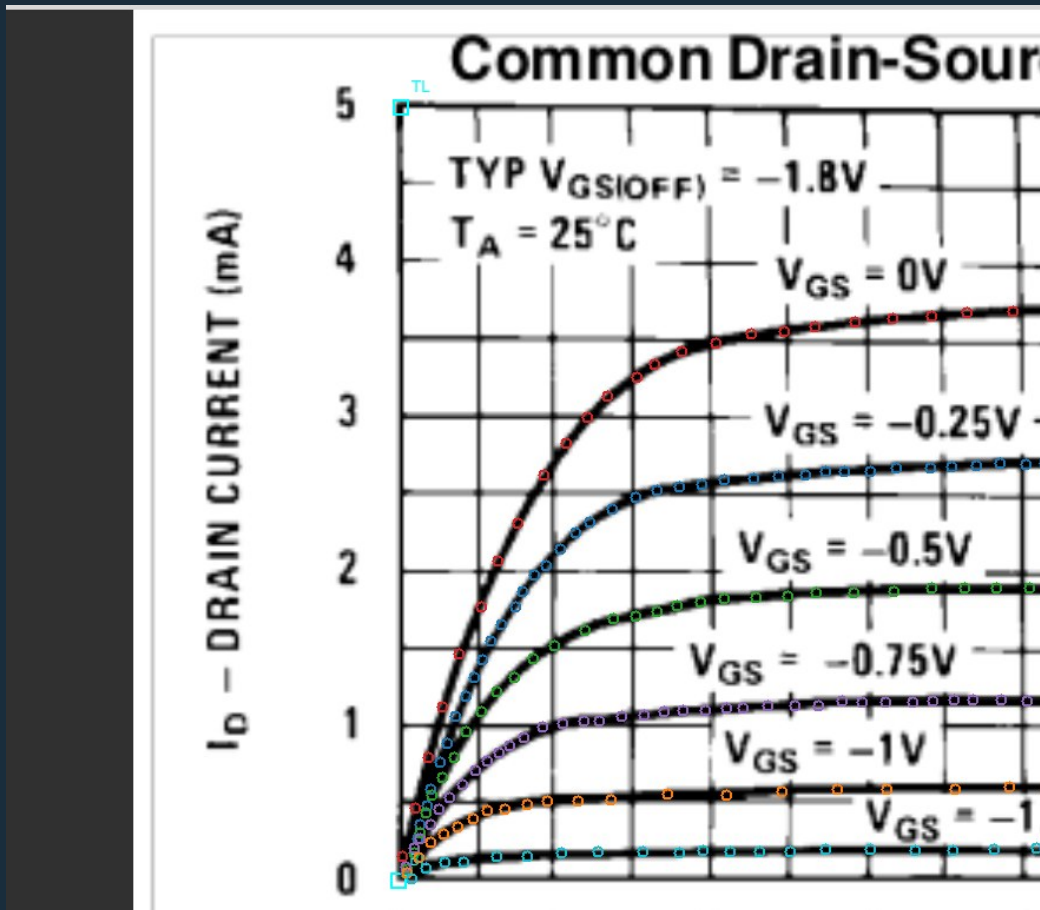


GX-TXT

Curve Digitizer

Complete Illustrated User Manual

Version **1.2.17** · Two complete real-interface workflows, followed by the full installed-viewer reference.



Electronic datasheet: six measured characteristics · Synthetic SEM micrograph: calibrated contours and non-geometric image processing

Contents and reading guide

The first two parts are **complete end-to-end procedures**. Follow the screenshots in order: they document actual steps performed in GX-TXT, from importing source files to publishing usable canonical Datasets. Every supplied screenshot is integrated at the relevant step. The final part reproduces the complete installed **MANUAL.md** in readable PDF typography, with its tables, operations, controls and developer-reference examples.

Contents and reading guide	2
PART I — Digitizing an electronic datasheet	4
1. Import the source into GX-TXT	4
2. Open the Digitizer and bind its source	5
3. Define the two scientific axes	5
4. Capture the four projective corners	6
5. Verify the calibration without changing it	8
6. Define the numerical family	8
7. Trace the individual family curves	9
8. Correct wrong clicks, do not restart tracing	10
9. Inspect the six completed traces	11
10. Make repeated projective calibrations	12
11. Choose a publication policy: separate or combined	14
12. Add the scientific notes	15
13. Publish the main combined family Dataset	16
14. Verify the published table and attached notes in GX-TXT	18
15. Render the result using Dataset Preview	19
PART II — SEM micrograph and image processing	21
1. Import and preview the SEM raster	21
2. Auto-calibrate the entire photograph in pixels	23

3. Detect the 10 μm scale bar	23
4. Convert the raster to physical coordinates	24
5. Verify a known micrometric position	25
6. Compare original and processed images	26
7. Inspect the actual processing order in Advanced mode	27
8. Publish an independent processed Image Dataset	28
9. Use the processed raster for profile digitization	30
PART III — Detailed viewer reference	30
GX-TXT central module configuration	31
Viewer settings: Display	31
Viewer settings: Digitizing	32
Viewer settings: Calibration	33
Viewer settings: Axes and output	34
Viewer settings: Dataset browser	35
Viewer settings: Configurations	36
APPENDIX A — Full installed viewer manual	37
APPENDIX B — Complete public API specifications	62
CALIBRATION API v1.md	62
CURVE DATASET SCHEMA API v1.md	69
IMAGE PROCESSING API v1.md	75

PART I — Digitizing an electronic datasheet

This worked example digitizes the six **common drain-source characteristics** of a 2N2222 device. The figure contains drain current as a function of drain-source voltage for several gate-source values. The result is a numerical curve-family Dataset usable by normal GX-TXT analysis and plotting modules. The same original pixels are also used for repeated projective calibrations and alternate Dataset publications.

1. Import the source into GX-TXT

Open or create a GX-TXT Session first. In the framework Acquisition workspace, choose the image-acquisition tool, select the PNG raster from the datasheet and import it as a canonical **image.raster** Dataset. The image must be registered in the Session before the Curve Digitizer can discover it; simply opening a file from the viewer is not the intended workflow.

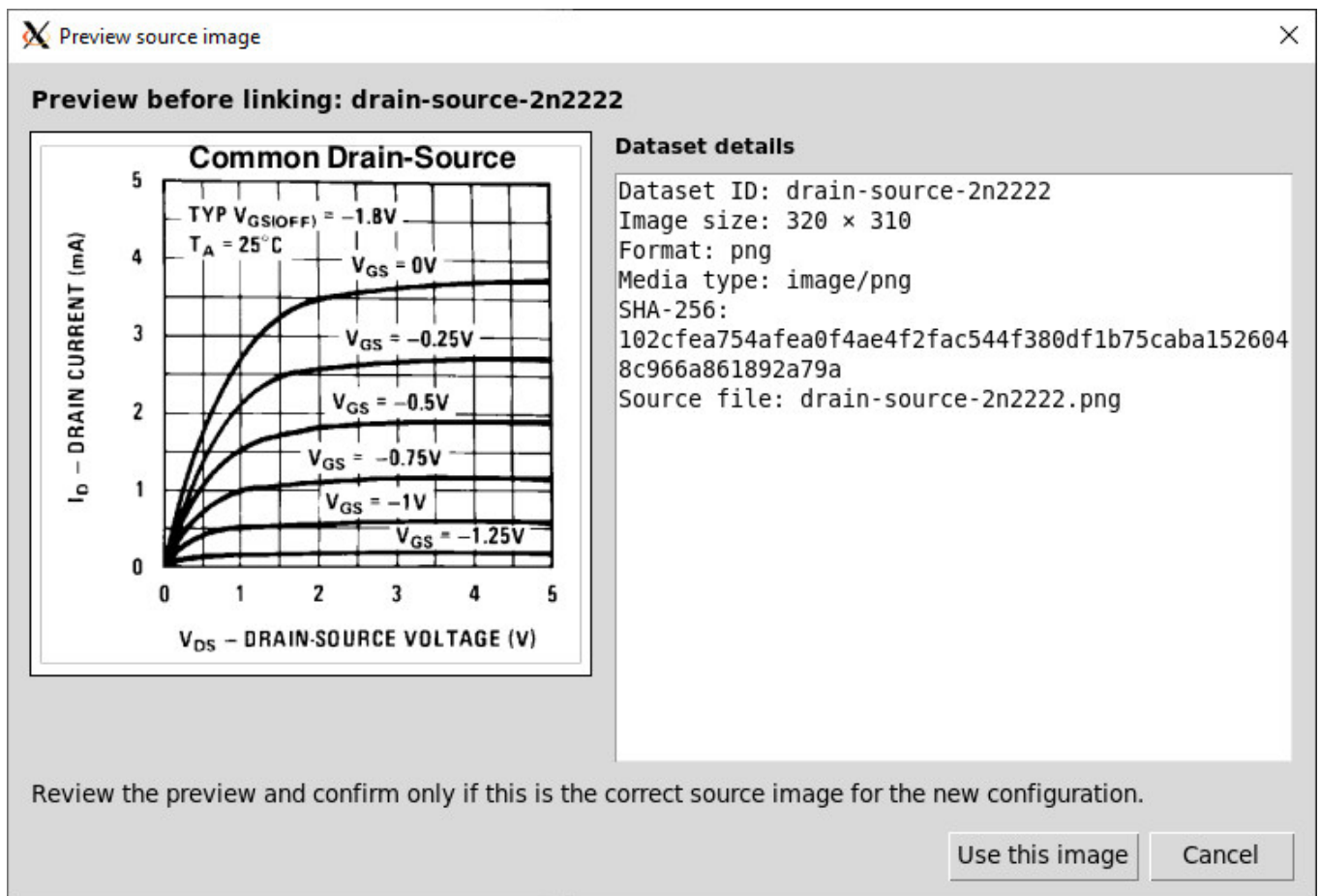


Figure 1. Acquisition import preview of the datasheet graph. Confirm the image, its raster dimensions and source-file identity before completing the import.

In this example the Dataset is named **drain-source-2n2222**. The acquisition preview demonstrates that GX-TXT has decoded the correct graph before it becomes the Digitizer source.

2. Open the Digitizer and bind its source

Choose **Viewers** → **Curve Digitizer**. Create a configuration and choose the previously imported image from **Image Dataset**. The configuration contains the curve points, calibration measurements, axis definitions and notes. Keep the selected source tied to its Dataset ID and SHA-256: a later change to image bytes must not be silently treated as the same geometric reference.

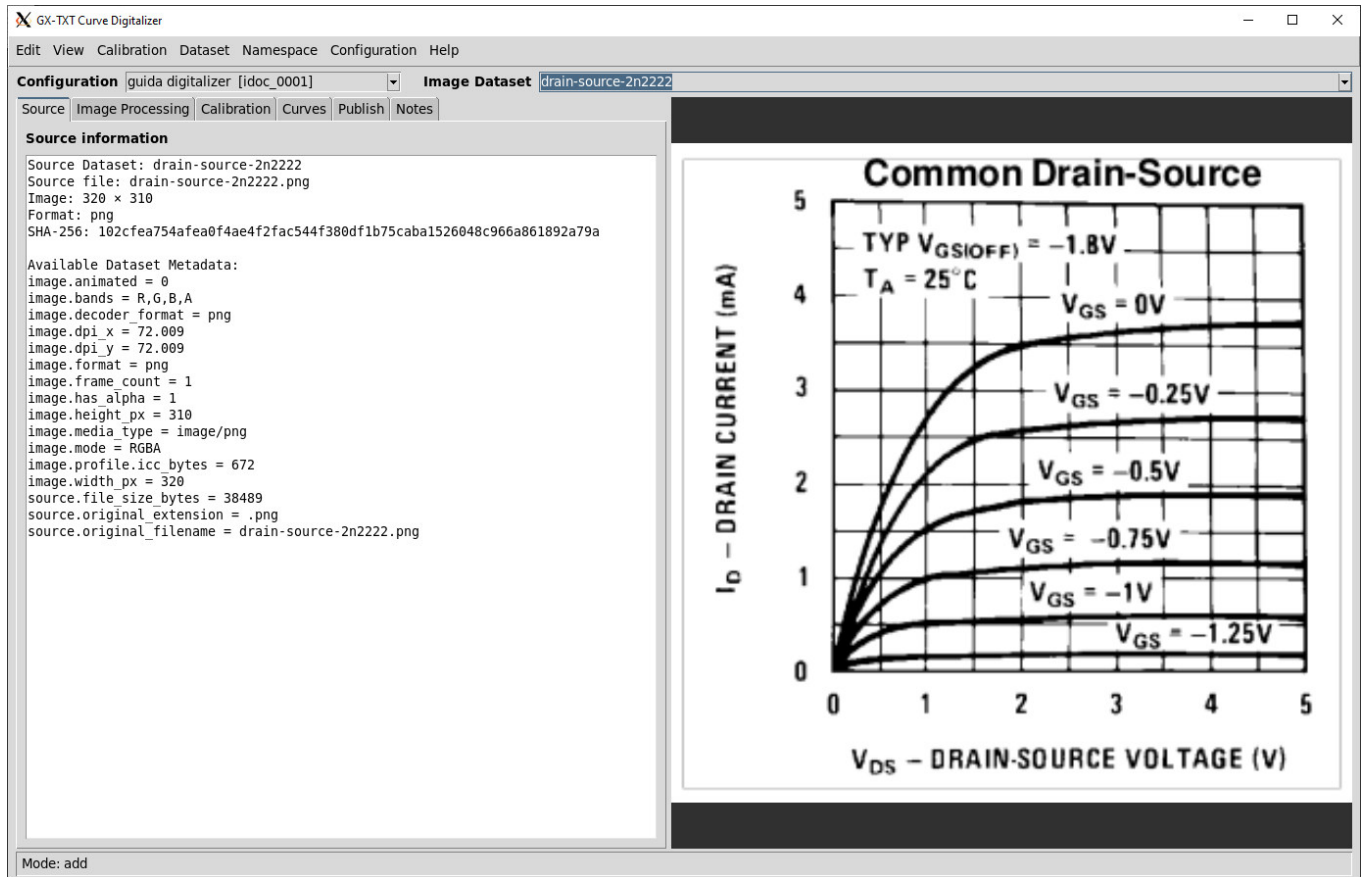
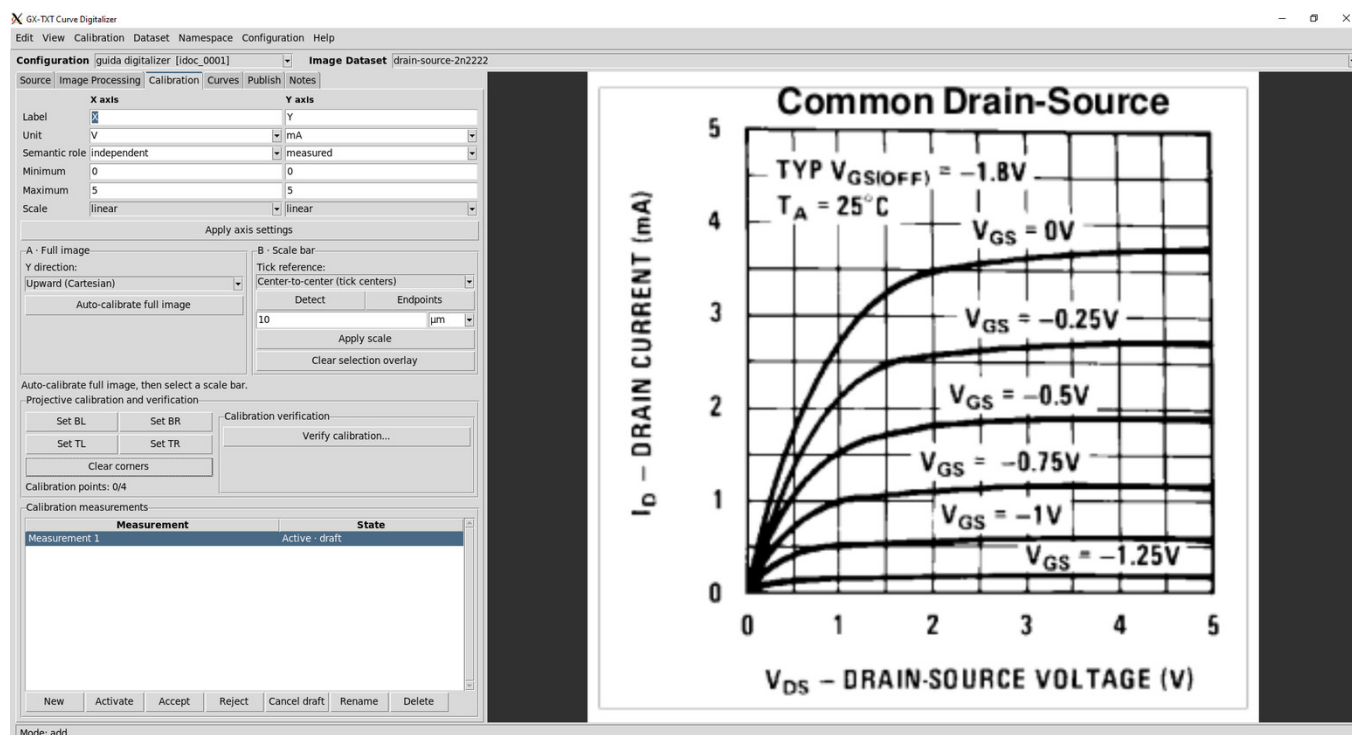


Figure 2. Source tab after linking the 2N2222 raster to a persistent Digitizer configuration. The configuration and image Dataset appear in the header.

3. Define the two scientific axes

Enter the physical quantities printed on the graph: **X = V_{DS} (V)** and **Y = I_D (mA)**. Both axes are linear in this example; their minimum is 0 and maximum is 5. Keep the semantic roles independent (X) and measured (Y). Confirm with **Apply axis settings** before placing the projective-corner markers.



Calibration and printed axes

4. Capture the four projective corners

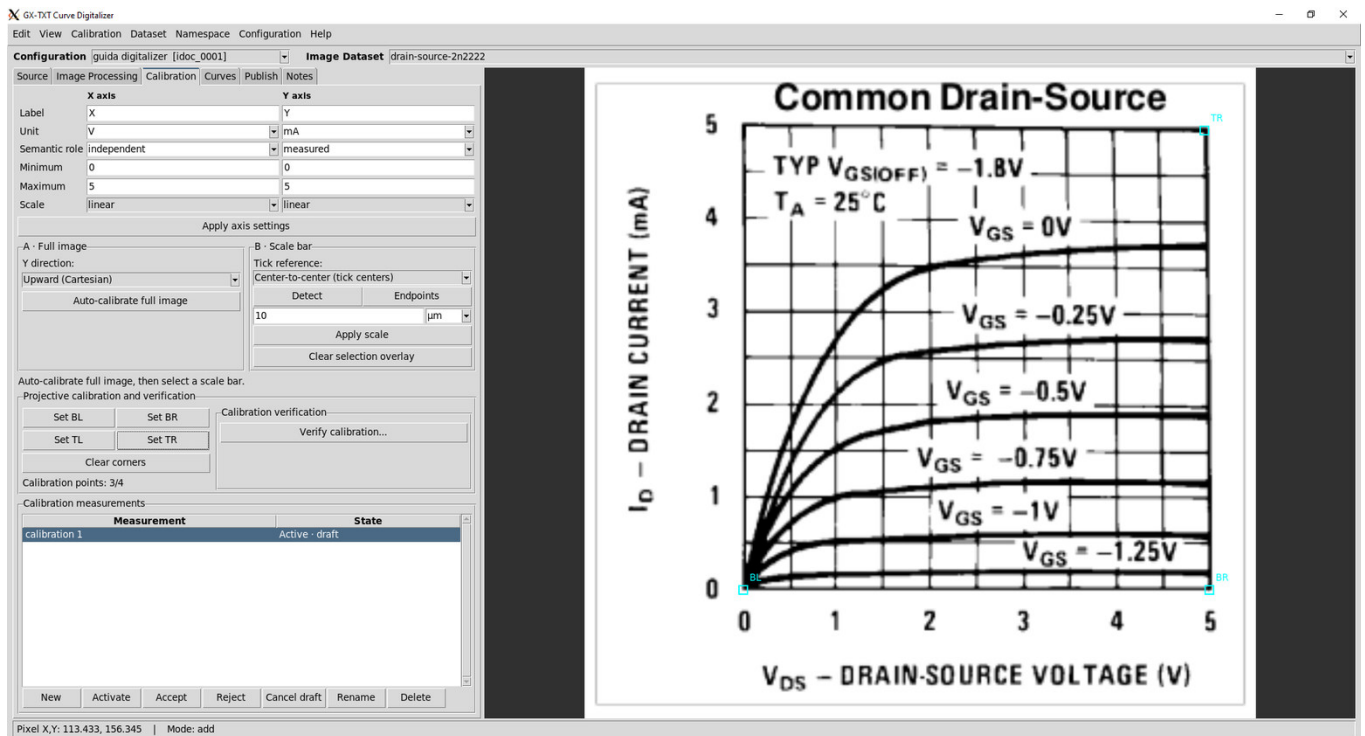


Figure 4. Projective calibration while the graph corners are still being selected. Incomplete calibration markers are visible.

Finish with all four points. The viewer displays a ready state only when the transformation is geometrically valid; readiness does not replace an independent check.

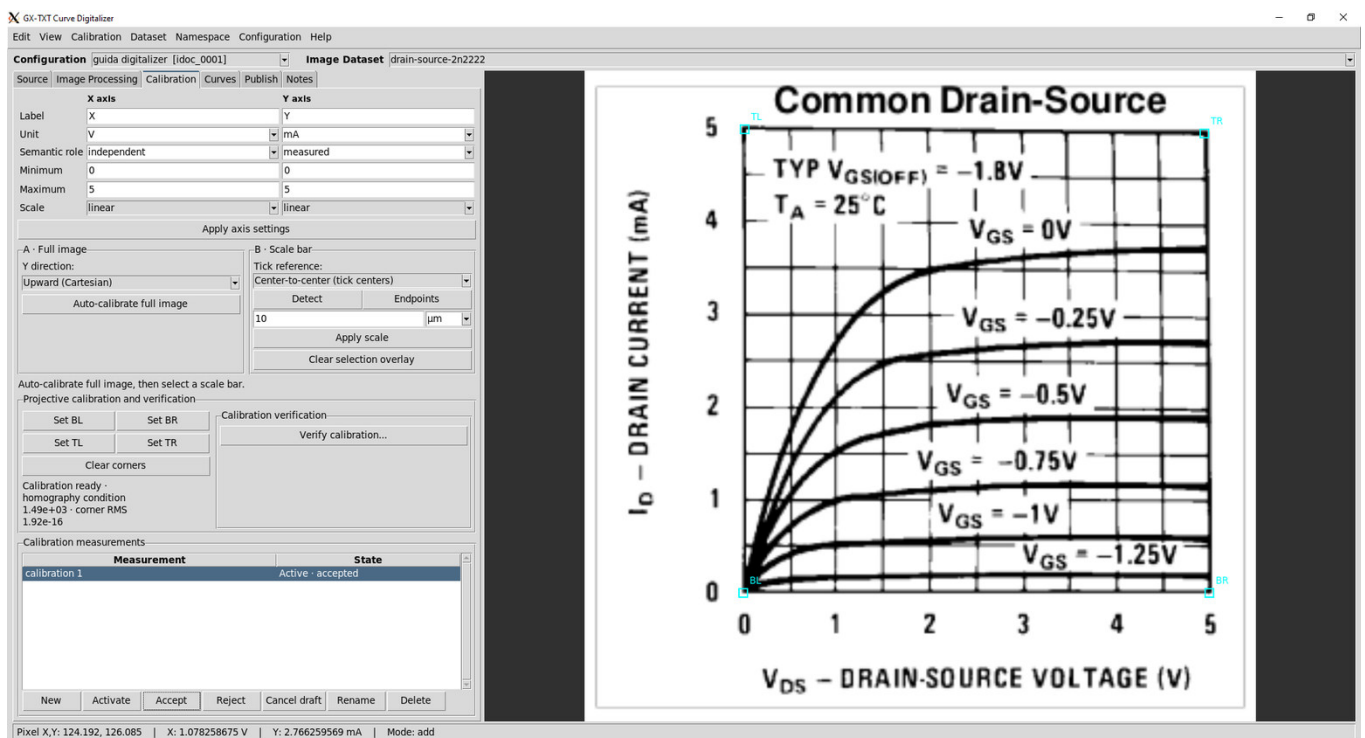


Figure 5. Projective calibration completed with all four corners set on the datasheet graph.

5. Verify the calibration without changing it

Open **Verify calibration...**, click an intersection of known grid coordinates and enter **Expected X** and **Expected Y**. The temporary red X marks the exact raster pixel being evaluated. The viewer reports the calibrated coordinates and their deviations from the expected values. You can edit the expected values after clicking and observe the residuals update.

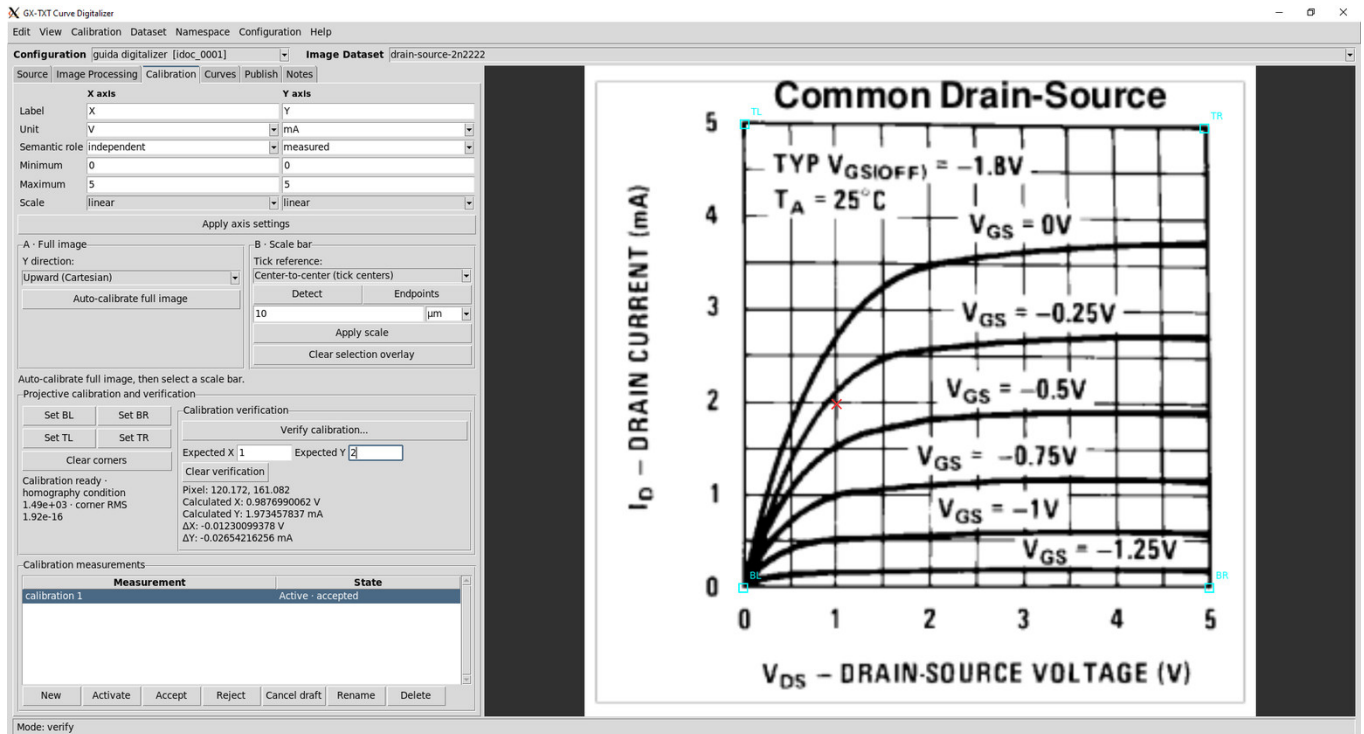


Figure 6. Independent calibration verification with a temporary red X and numerical X/Y residuals.

Use this residual as a validation observation, not as an automatically established measurement uncertainty. Press Esc or leave verification mode to clear the temporary marker.

6. Define the numerical family

Switch to **Curves** and enable **This configuration is a numerical curve family**. Set the family quantity to gate-source voltage V_{GS} , with unit V. The six curves share X and Y physics; they differ by numeric family values. Give every member a descriptive label and a value from the graph (0, -0.25, -0.50, -0.75, -1.00 or -1.25 V).

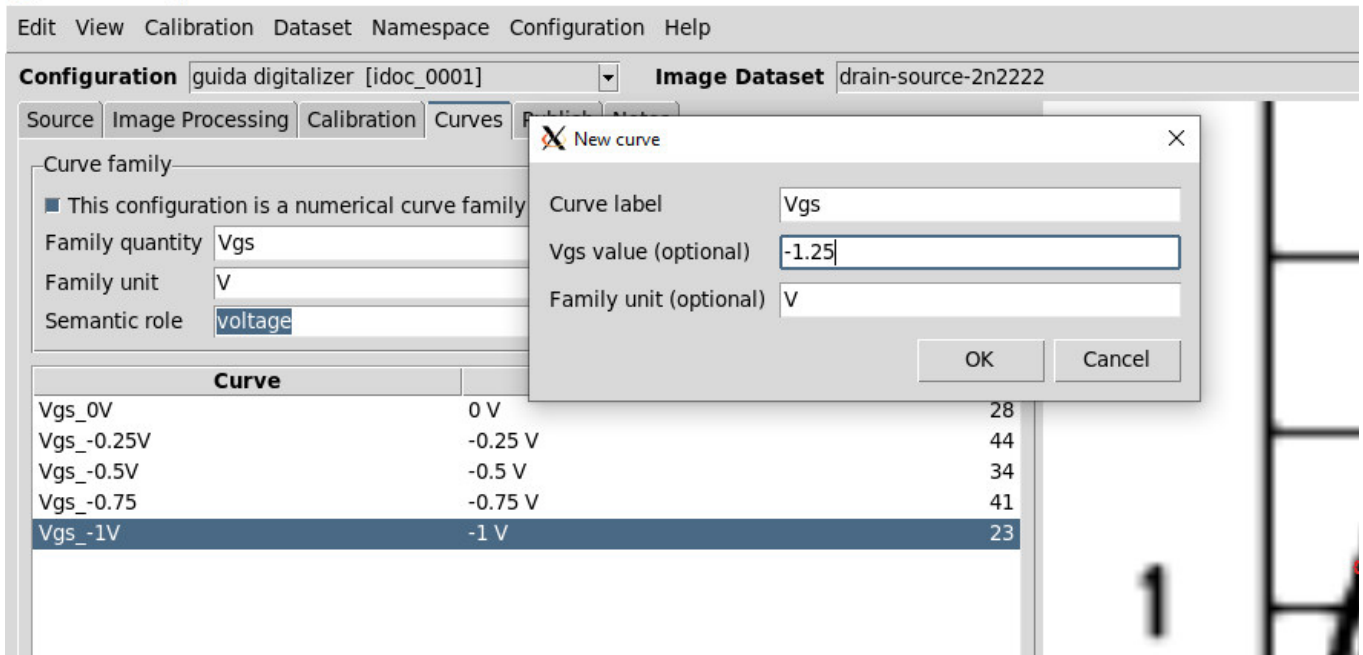


Figure 7. Family configuration and the new-curve dialog, showing the relationship between family settings and the value entered for a curve.

Family versus independent curves

Use the numerical family mode only when the curves genuinely differ by a numeric parameter. For arbitrary independent profiles, leave the family checkbox disabled to publish the correct non-family schema.

7. Trace the individual family curves

Select the intended member and activate **Add points**. Click the raster line successively in natural X order, increasing density where the curve has strong curvature. The viewer stores original raster positions, and recalculates scientific coordinates from the active calibration. Point modification is deliberately restricted to the **Curves** tab.

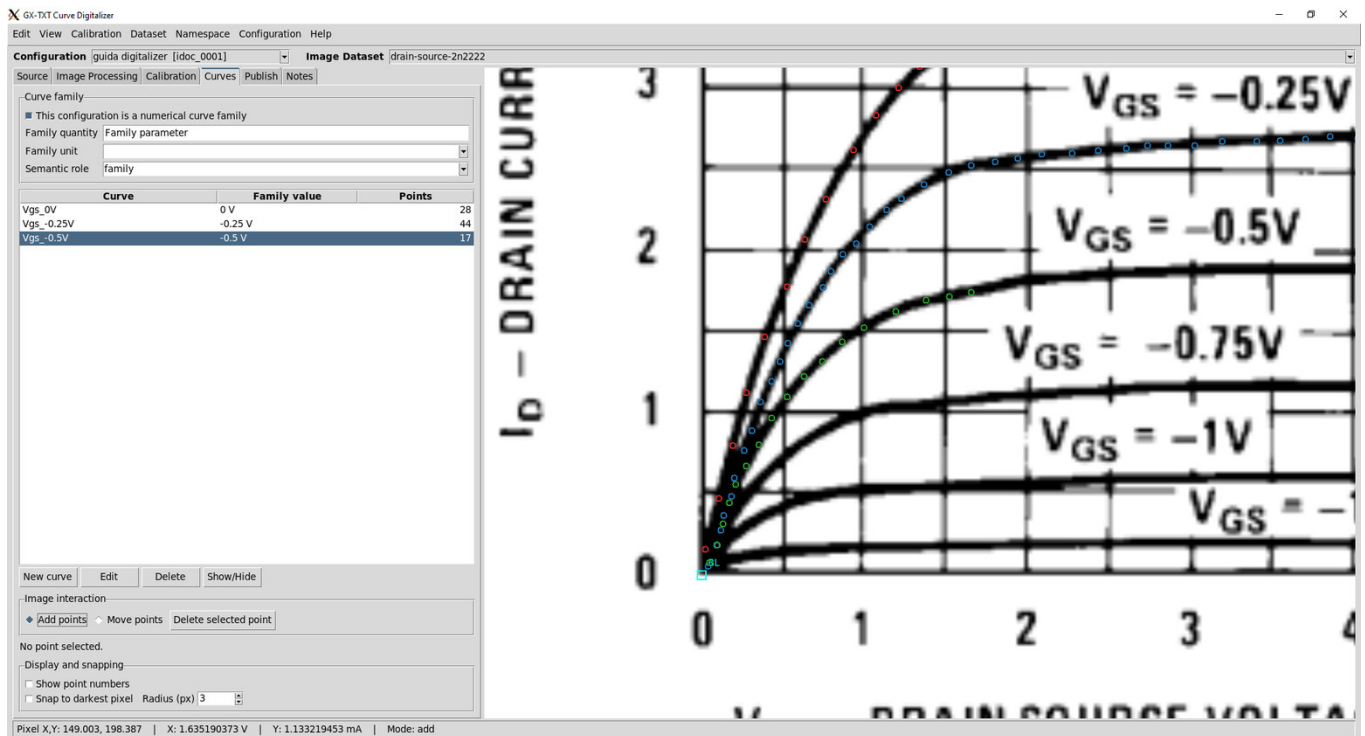


Figure 8. Progressive point acquisition along one of the drain-source curves. Each marker is tied to the currently selected member.

8. Correct wrong clicks, do not restart tracing

Raster plots frequently contain grid lines and thick printed strokes. A point can accidentally snap to the wrong feature. Inspect every curve and use **Move points** to select and move an incorrect point; remove it only when necessary. Undo and Redo protect ordinary editing.

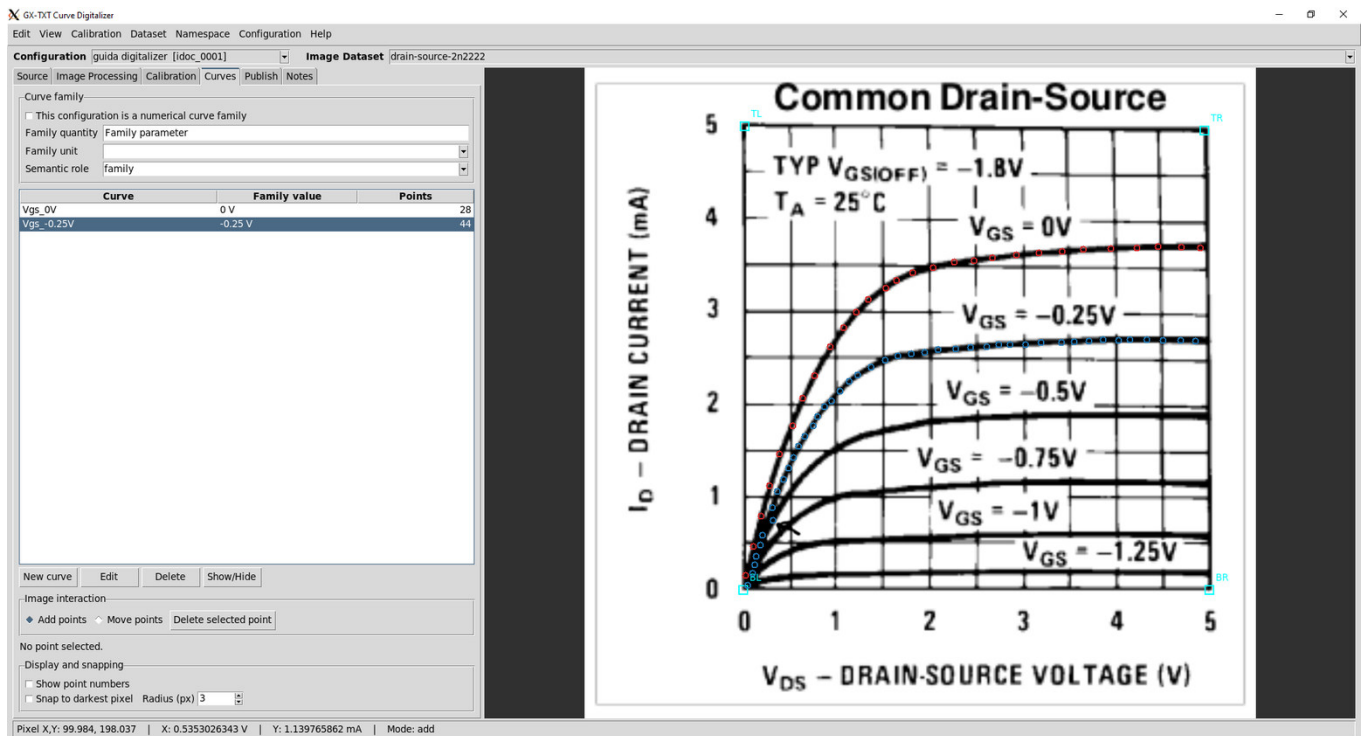


Figure 9. A misplaced sample point: the selected raster position does not follow the intended curve.

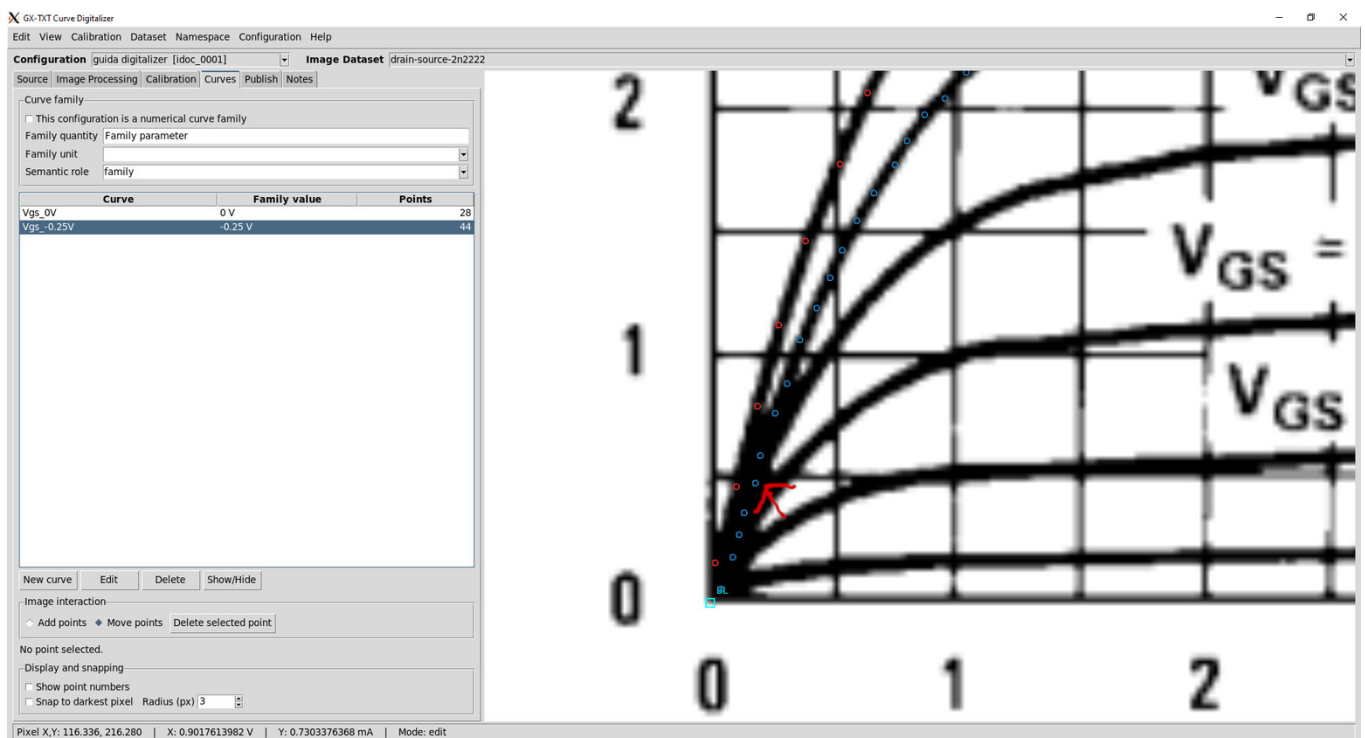


Figure 10. The same part of the source image after moving the incorrect point back onto its intended trace.

9. Inspect the six completed traces

Once all members are digitized, show them together. Their overlay provides a visual check for omitted regions, swapped family values and coarse sampling. This example yields six named curves and 193 source rows in the published family.

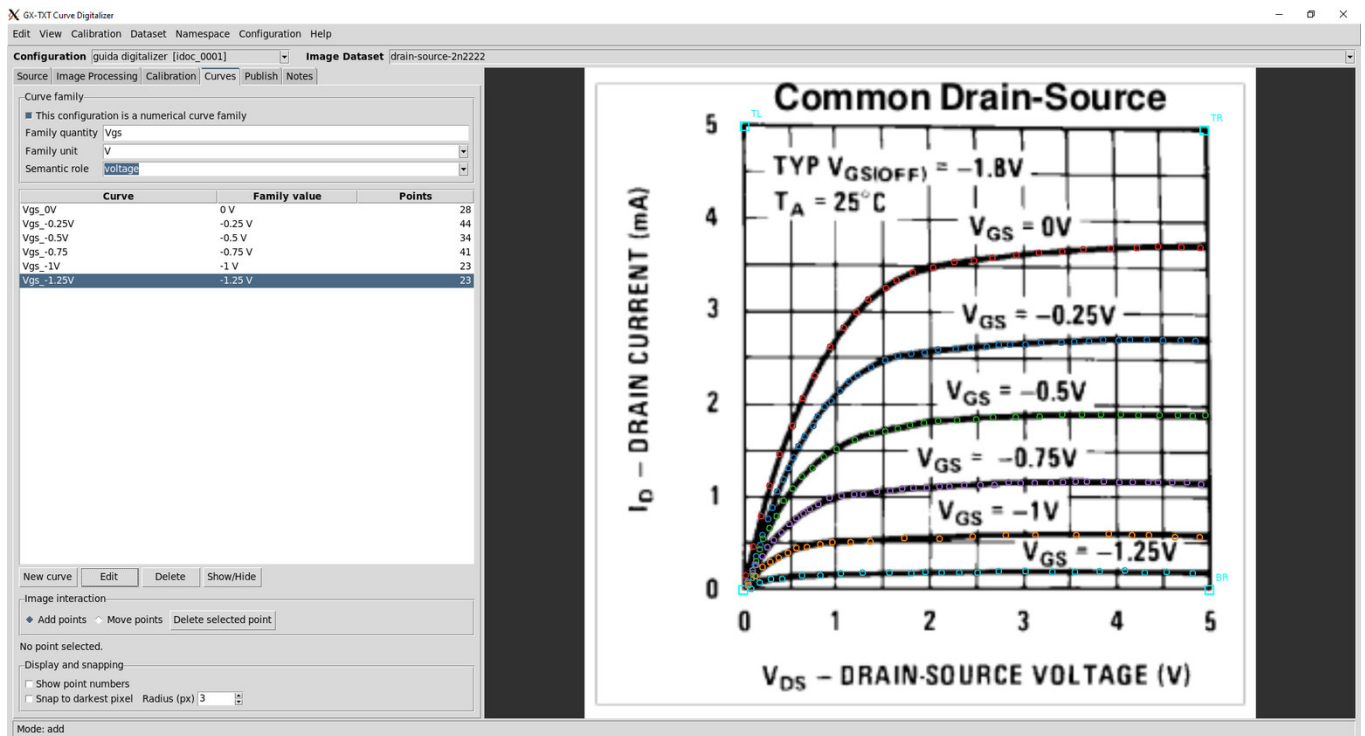


Figure 11. Completed 2N2222 curve family. All digitized members are visible over the raster graph.

10. Make repeated projective calibrations

A calibration measurement can be repeated without touching the acquired curve points. In **Calibration measurements**, create and accept another set of four corner selections. Each accepted measurement has a persistent ID, a label, an independent transformation and a reproducible fingerprint. Activate whichever one should drive the on-screen scientific coordinates.

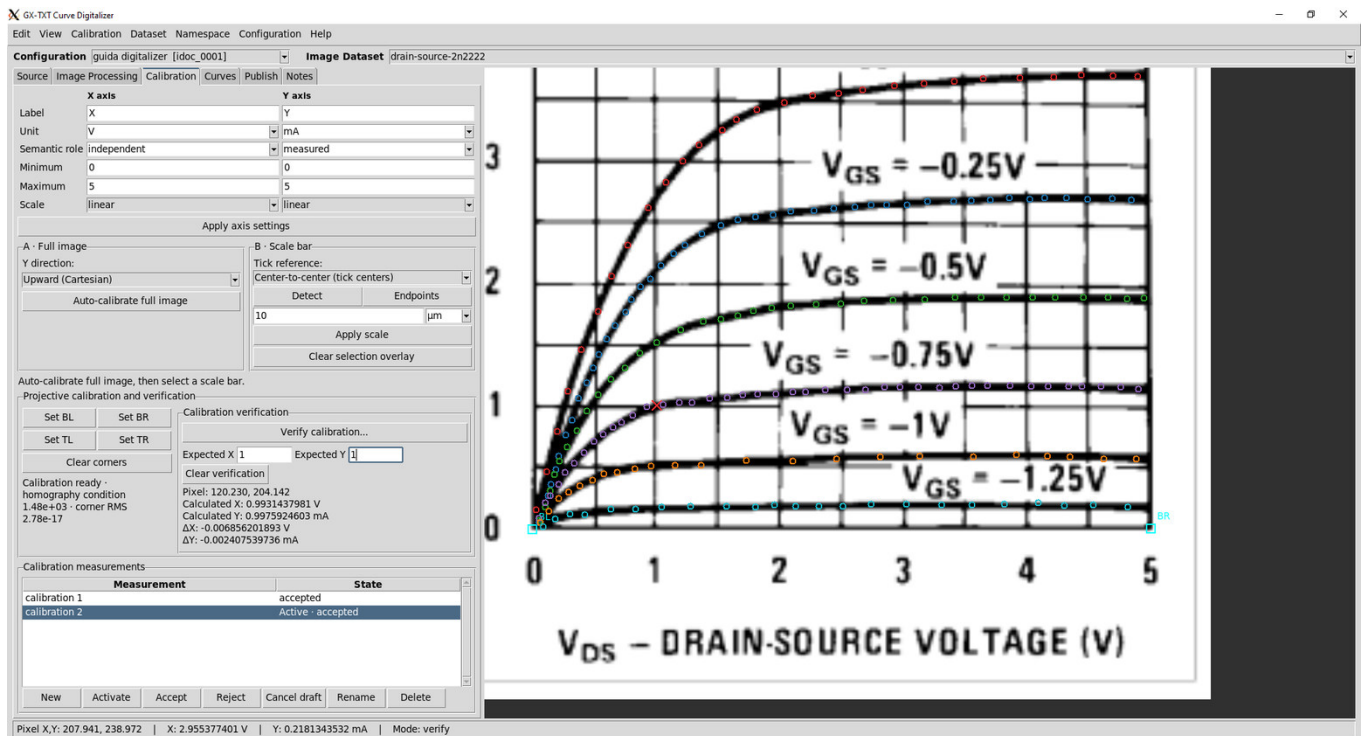


Figure 12. Second projective calibration of the same original datasheet figure, using the same traced raster points.

Repeat once more and use independent verification on the resulting transformation. Multiple realizations expose the variability associated with manual corner placement; the viewer does not turn that variability into an invented uncertainty estimate.

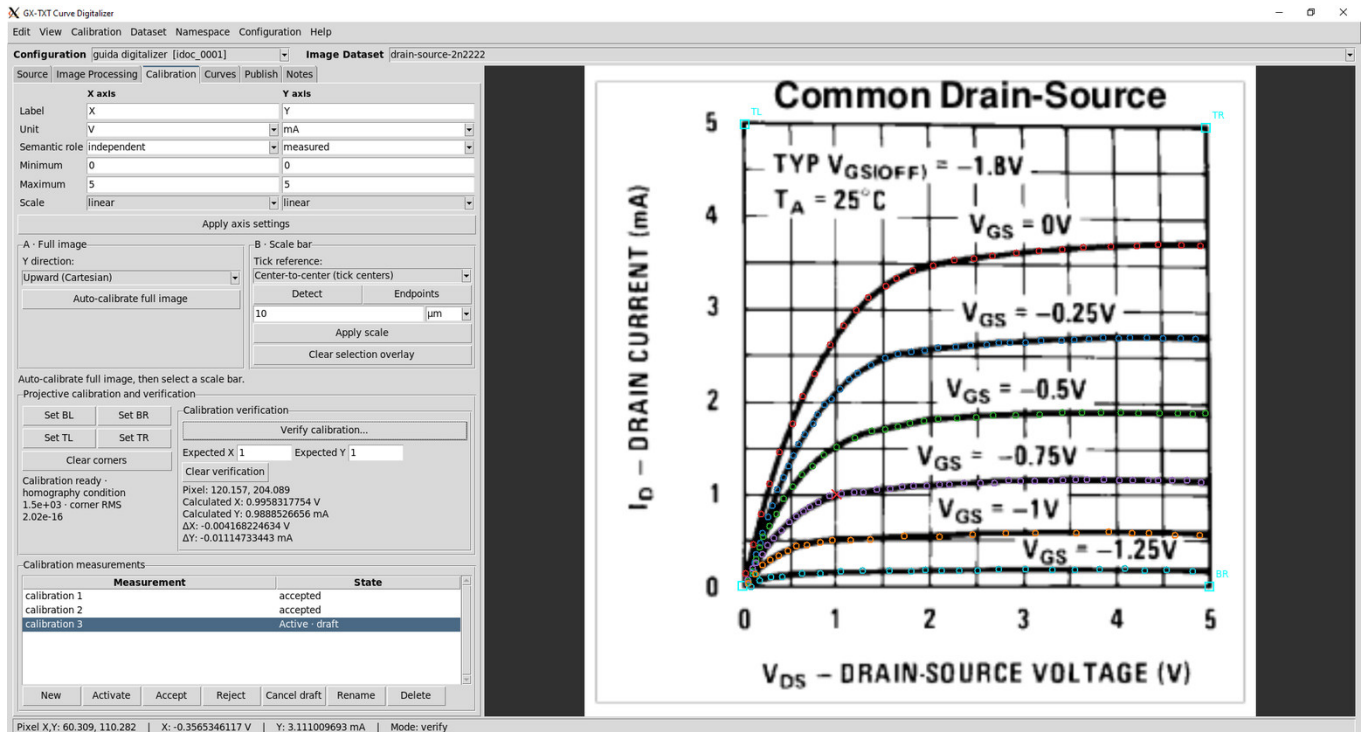


Figure 13. Third calibration on the 2N2222 graph with the corresponding verification result.

11. Choose a publication policy: separate or combined

The **Publish** tab lists accepted calibration measurements. Select one or more explicitly; the active calibration is not necessarily the only one published. Choose **Separate** when a consumer expects one conventional curve-family Dataset per calibration. A naming template such as **[name].[calibration].curves** produces stable, distinguishable Dataset IDs.

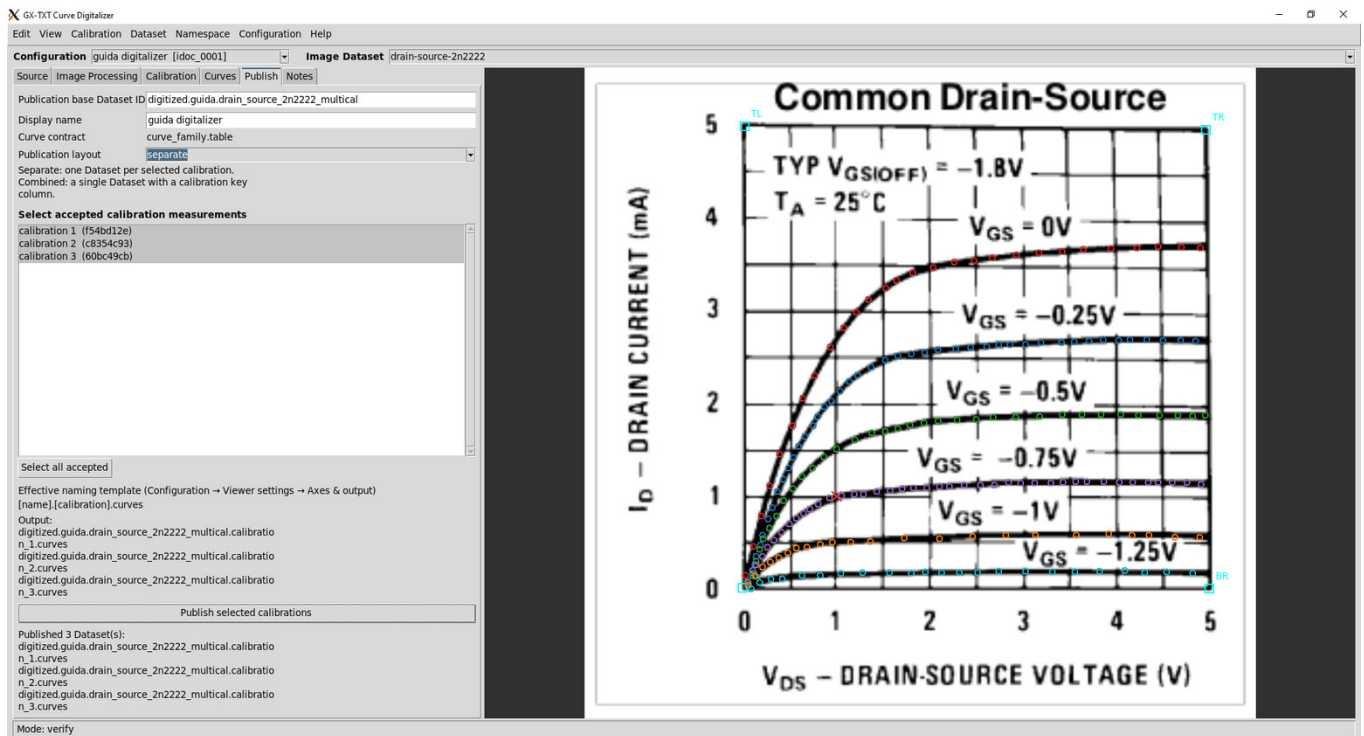


Figure 14. The publication panel in Separate mode, with three selected accepted calibrations and the resulting ID preview.

After publication, verify that GX-TXT contains distinct Datasets, one for each selected calibration. Each Dataset retains the same raster points but scientific values transformed by the corresponding calibration.

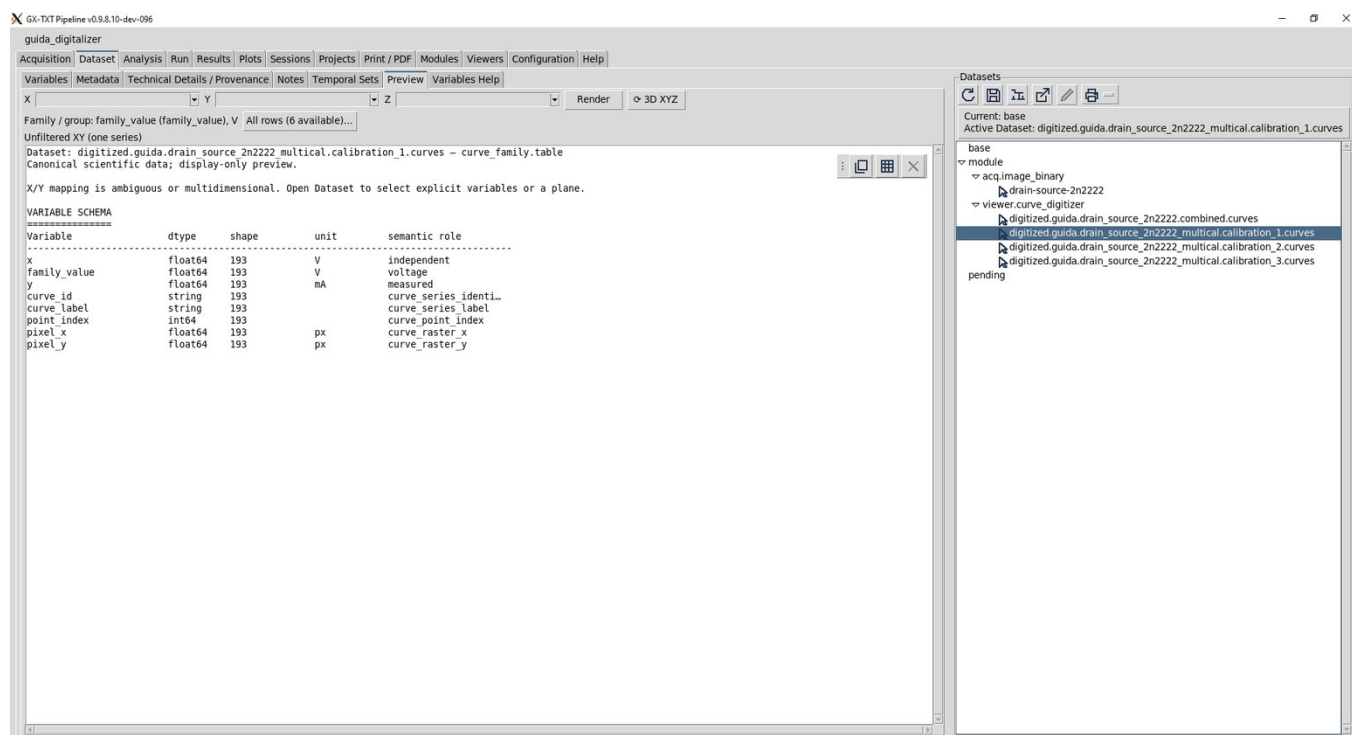


Figure 15. GX-TXT Dataset browser showing the separate scientific Datasets created from multiple accepted calibrations.

Combined is the other supported option: all chosen calibrations share one Dataset and are distinguished by explicit calibration-key/name semantics. Prefer Separate mode if a downstream consumer does not understand the extra calibration dimension.

12. Add the scientific notes

Use **Notes** to document the component, datasheet figure, family parameter and conditions printed in the source. Notes are human-readable scientific context, not a replacement for structured calibration metadata. On publication, the viewer adds them to the canonical Dataset note document in GX-TXT.

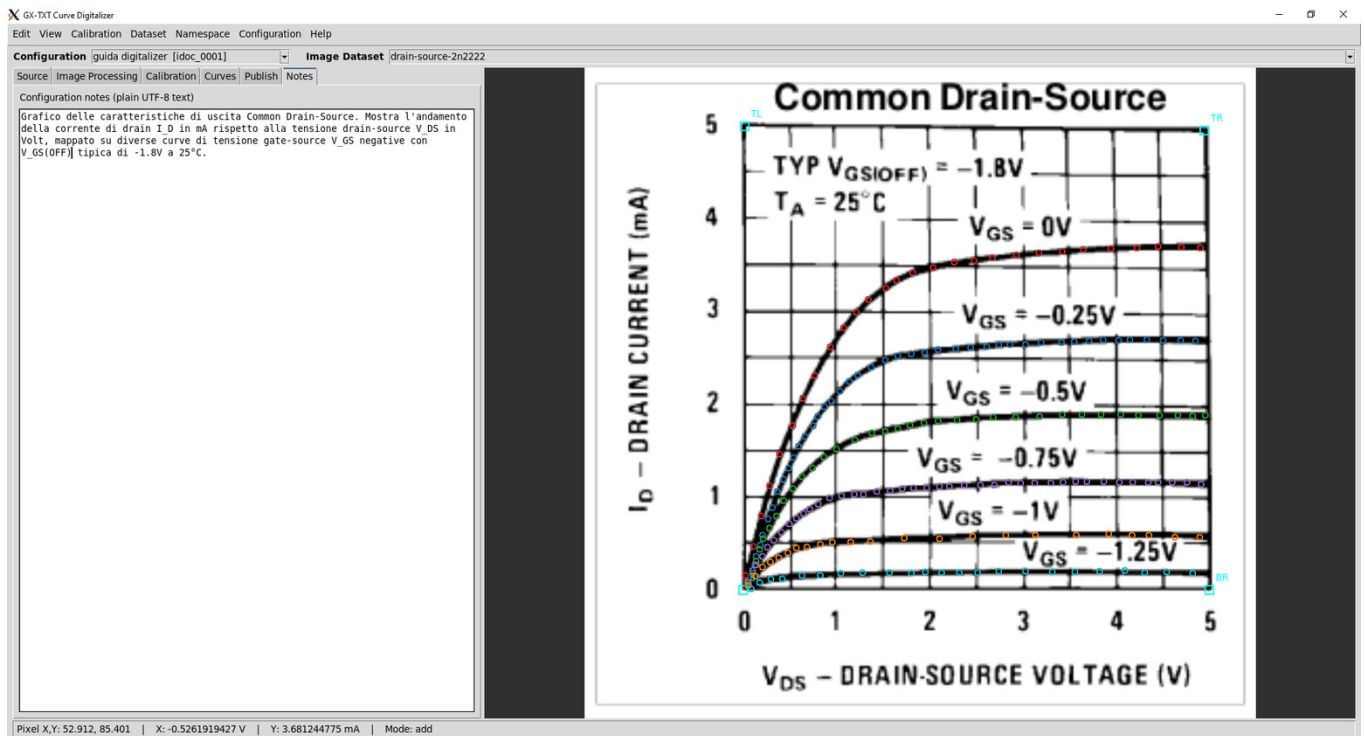


Figure 16. The Notes tab with the source conditions and remarks recorded for the family digitization.

13. Publish the main combined family Dataset

For the illustrated primary result, choose the **curve_family.table** contract and the **Combined** publication layout. Verify the requested Dataset ID and calibration selection. A combined family can still be plotted as six numerical members with the family grouping metadata. Confirm the previewed identity before publishing.

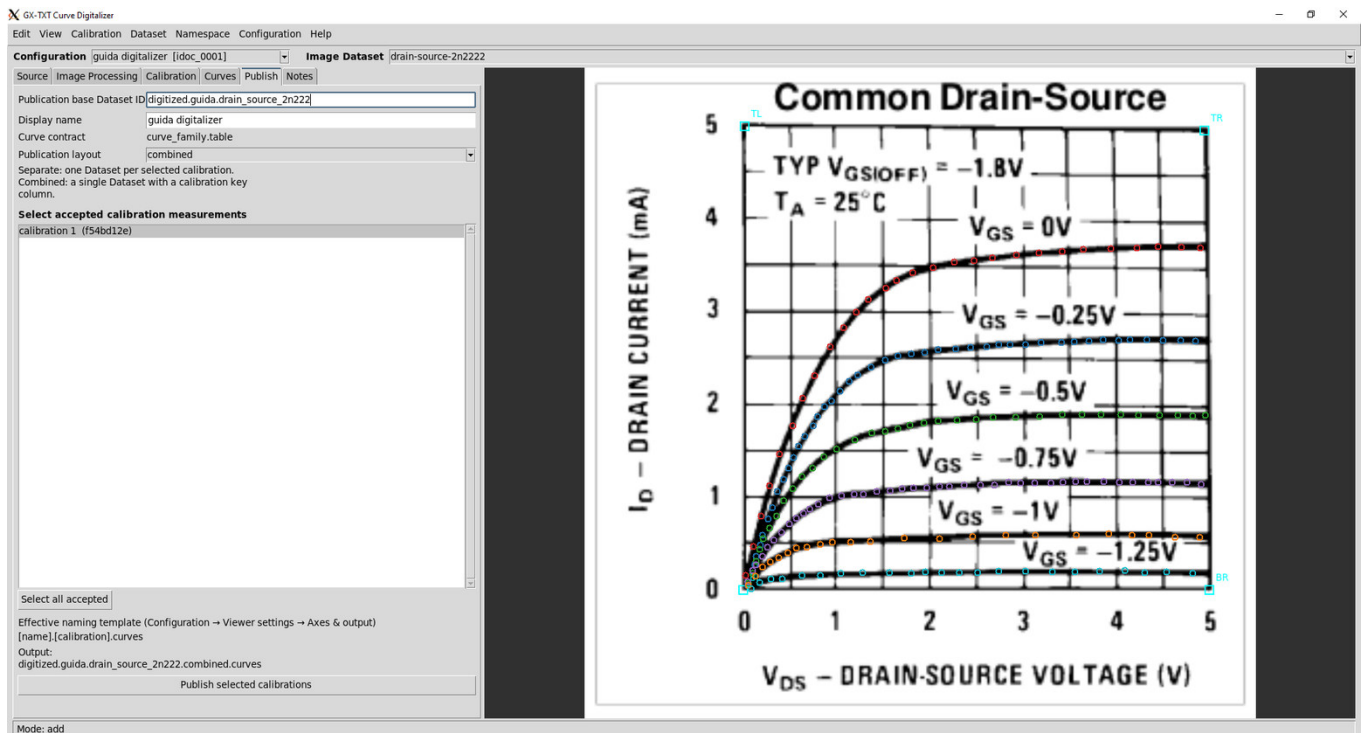


Figure 17. Publication settings for the canonical combined curve-family Dataset, including identity and output contract.

Click **Publish selected calibrations**. The viewer creates a canonical Dataset; it does not merely save an image or export unregistered CSV bytes. A successful result reports the actual published Dataset identity.

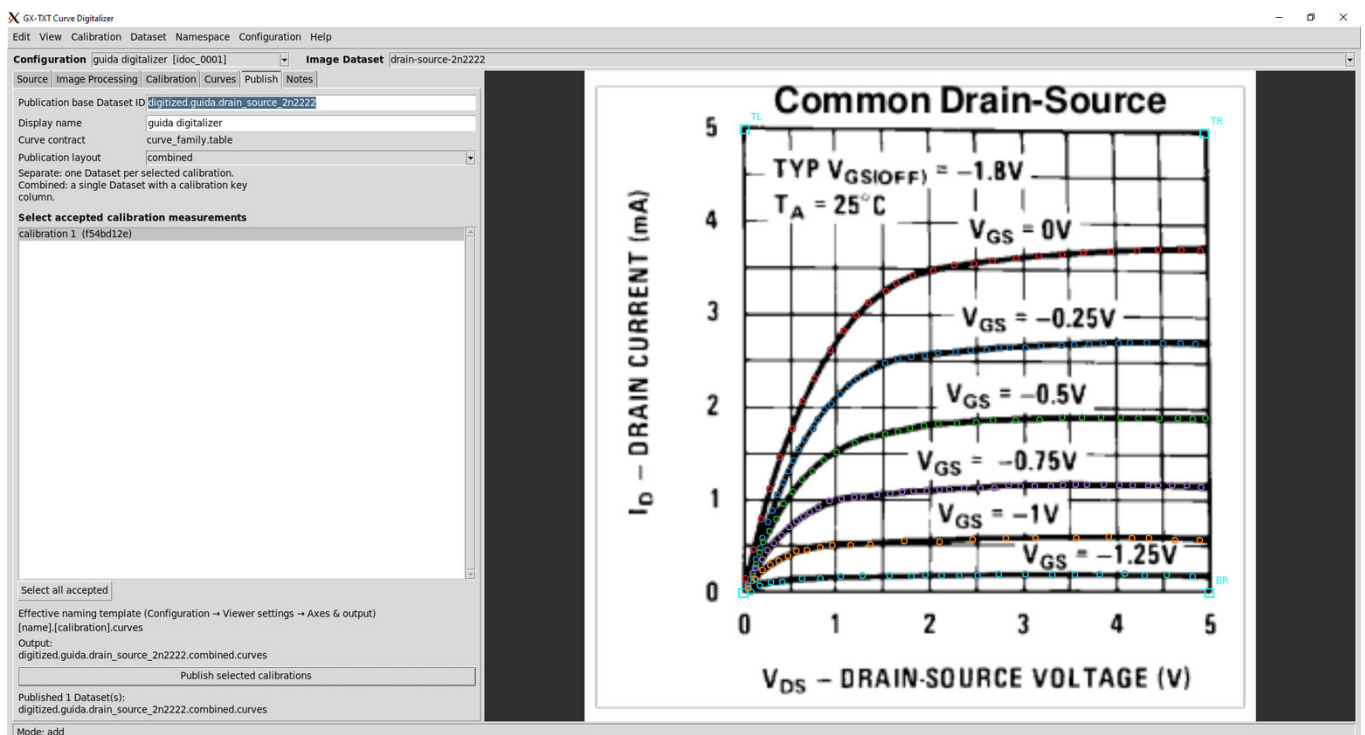


Figure 18. Publish operation completed and the new Dataset identity reported by Curve Digitizer.

14. Verify the published table and attached notes in GX-TXT

Open the new Dataset from the GX-TXT Dataset browser. Verify that it contains numeric coordinates, family values, curve identities, point ordering and original raster columns. A consumer should resolve roles through the public Dataset Schema API rather than depending on literal column names.

The screenshot shows the GX-TXT software interface. The main window displays a table with columns: Source row, X (V), Vgs (V), Y (mA), Curve ID, Curve label, and Point in. The table contains 193 rows of data. A 'VARIABLE SCHEMA' window is open, showing the dataset's structure with columns: Variable, dtype, shape, unit, and semantic role. The schema lists variables like family_value, curve_id, curve_label, point_index, calibration_id, calibration_label, pixel_x, and pixel_y with their respective data types and units.

Source row	X (V)	Vgs (V)	Y (mA)	Curve ID	Curve label	Point in
1	0.0200455262701205	0	0.1604903062872081	bfebbe2-c449-4ff2-b	Vgs_OV	0
2	0.1030924042988624	0	0.4668699055615228	bfebbe2-c449-4ff2-b	Vgs_OV	1
3	0.1862707860511793	0	0.7887166004162524	bfebbe2-c449-4ff2-b	Vgs_OV	2
4	0.2696094125510700	0	1.1161845419256293	bfebbe2-c449-4ff2-b	Vgs_OV	3
5	0.3778528137364627	0	1.4543250989233227	bfebbe2-c449-4ff2-b	Vgs_OV	4
6	0.5161155006645772	0	1.7633838790410654	bfebbe2-c449-4ff2-b	Vgs_OV	5
7	0.6249195114275963	0	2.0530522493831853	bfebbe2-c449-4ff2-b	Vgs_OV	6
8	0.7538381323181597	0	2.3033504823351363	bfebbe2-c449-4ff2-b	Vgs_OV	7
9	0.9227860796655491	0	2.6093872586174545	bfebbe2-c449-4ff2-b	Vgs_OV	8
10	1.0622028417787985	0	2.8206171488716629	bfebbe2-c449-4ff2-b	Vgs_OV	9
11	1.2018308444866845	0	2.992006635969217	bfebbe2-c449-4ff2-b	Vgs_OV	10
12	1.331643835062861	0	3.1233609339597623	bfebbe2-c449-4ff2-b	Vgs_OV	11
13	1.5213466384398155	0	3.2473297656031903	bfebbe2-c449-4ff2-b	Vgs_OV	12
14	1.6366644413221625	0	3.3364340698277166	bfebbe2-c449-4ff2-b	Vgs_OV	13
15	1.8069091371367572	0	3.4180564605395611	bfebbe2-c449-4ff2-b	Vgs_OV	14
16	2.0273771934520228	0	3.4748491715386653	bfebbe2-c449-4ff2-b	Vgs_OV	15
17	2.2581579574589976	0	3.5368207841911254	bfebbe2-c449-4ff2-b	Vgs_OV	16
18	2.4739919546493865	0	3.5533057166767001	bfebbe2-c449-4ff2-b	Vgs_OV	17
19	2.670002205978081	0	3.5848445015653088	bfebbe2-c449-4ff2-b	Vgs_OV	18
20	2.9316185872340437	0	3.6218853854638384	bfebbe2-c449-4ff2-b	Vgs_OV	19

Figure 19. Scientific table of the resulting combined Dataset in GX-TXT, with source rows and variable definitions.

Inspect its **Notes** tab as well: the originally entered text belongs to the published Dataset and is maintained as a Session Markdown note.

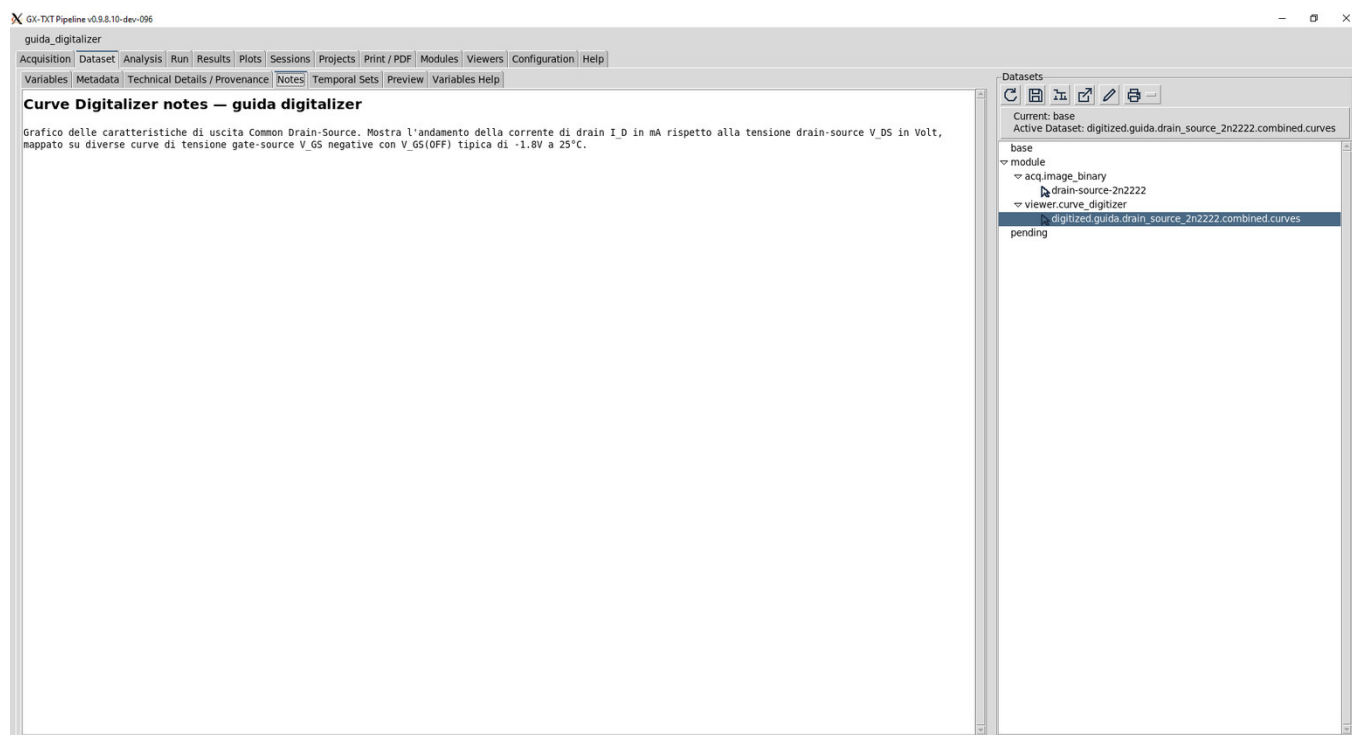


Figure 20. The text from Curve Digitizer Notes available from the published Dataset Notes interface.

15. Render the result using Dataset Preview

Without reopening Curve Digitizer, select the published Dataset and its X, Y and **family_value** bindings in the framework Dataset Preview. GX-TXT renders the six characteristic curves as ordinary scientific data. This demonstrates reuse of the published result beyond the producing viewer.

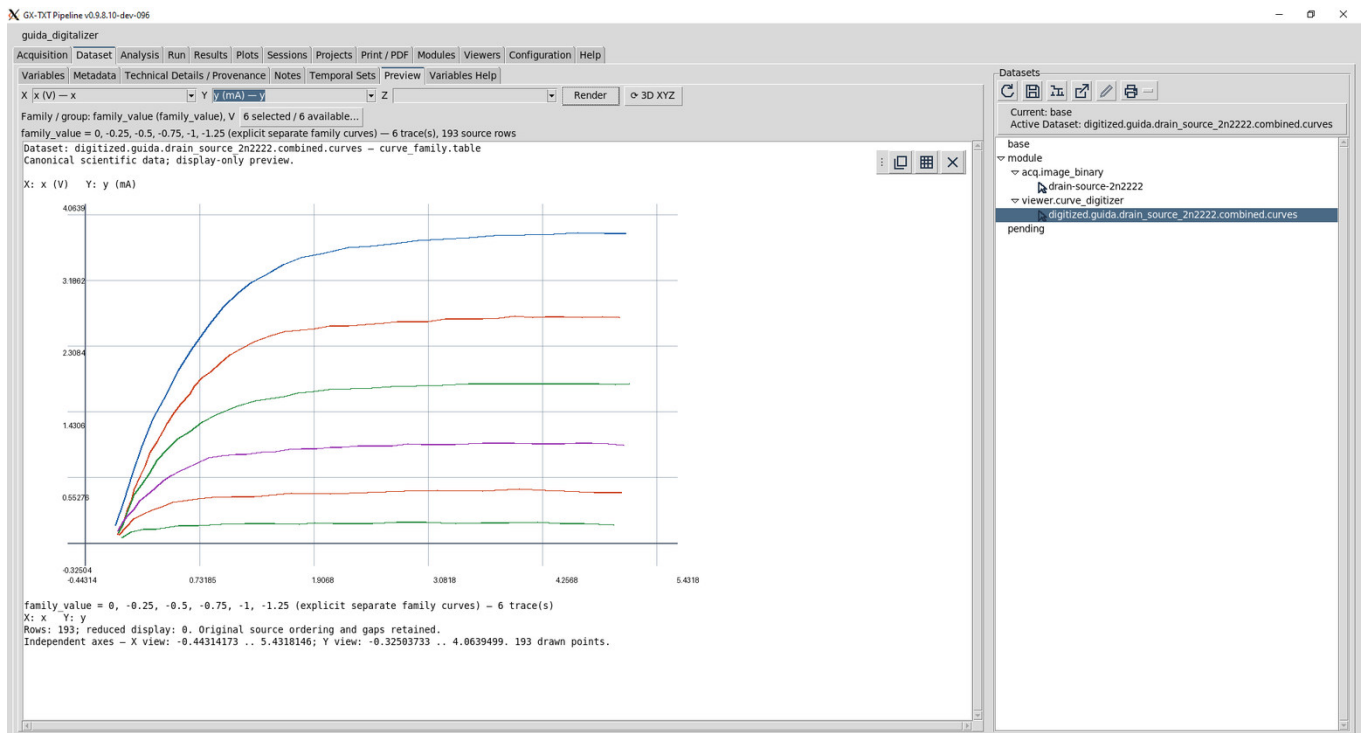


Figure 21. Dataset Preview rendering all six published family members and showing the canonical result in the Dataset catalog.

The original source plot is retained for comparison. The final extracted raster graph is shown below as an overview of the worked example.

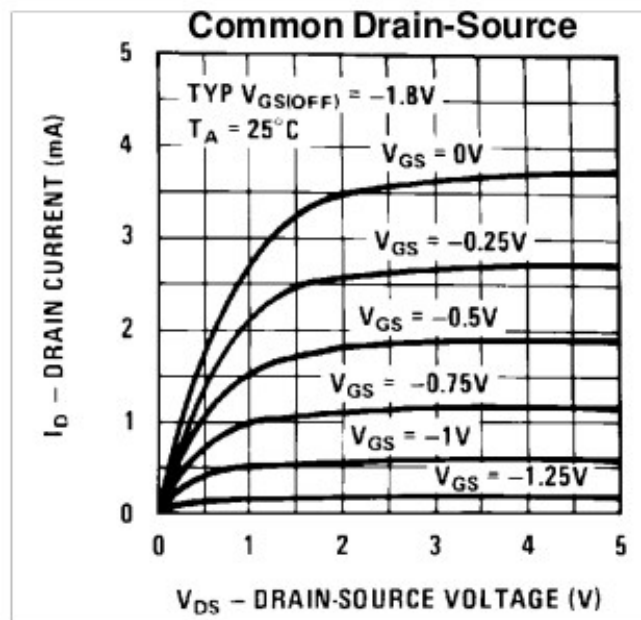


Figure 22. Source chart used in the datasheet tutorial, showing the six output characteristics selected for digitalization.

End-to-end verification

At this point the image has passed through GX-TXT import, source linking, physical axis calibration, curve sampling, optional calibration replication, Dataset publication, inspection of table/notes and independent plotted reuse.

PART II — SEM micrograph and image processing

The second procedure uses the supplied **synthetic SEM-style training image** of a metallic microfissure. It is suitable for teaching the interface, contour tracing and scale-bar procedures; it is **not a calibrated acquisition from a physical microscope** and must not be presented as real measurement evidence. The image includes two opposing irregular edges and an illustrative 10 µm graphic scale bar.

1. Import and preview the SEM raster

In GX-TXT choose the image importer, select the supplied PNG and confirm the preview, dimensions, media type and Dataset name. Do not start the Digitizer with an unregistered file. The imported raster becomes the authority for all subsequent calibration and derived products.

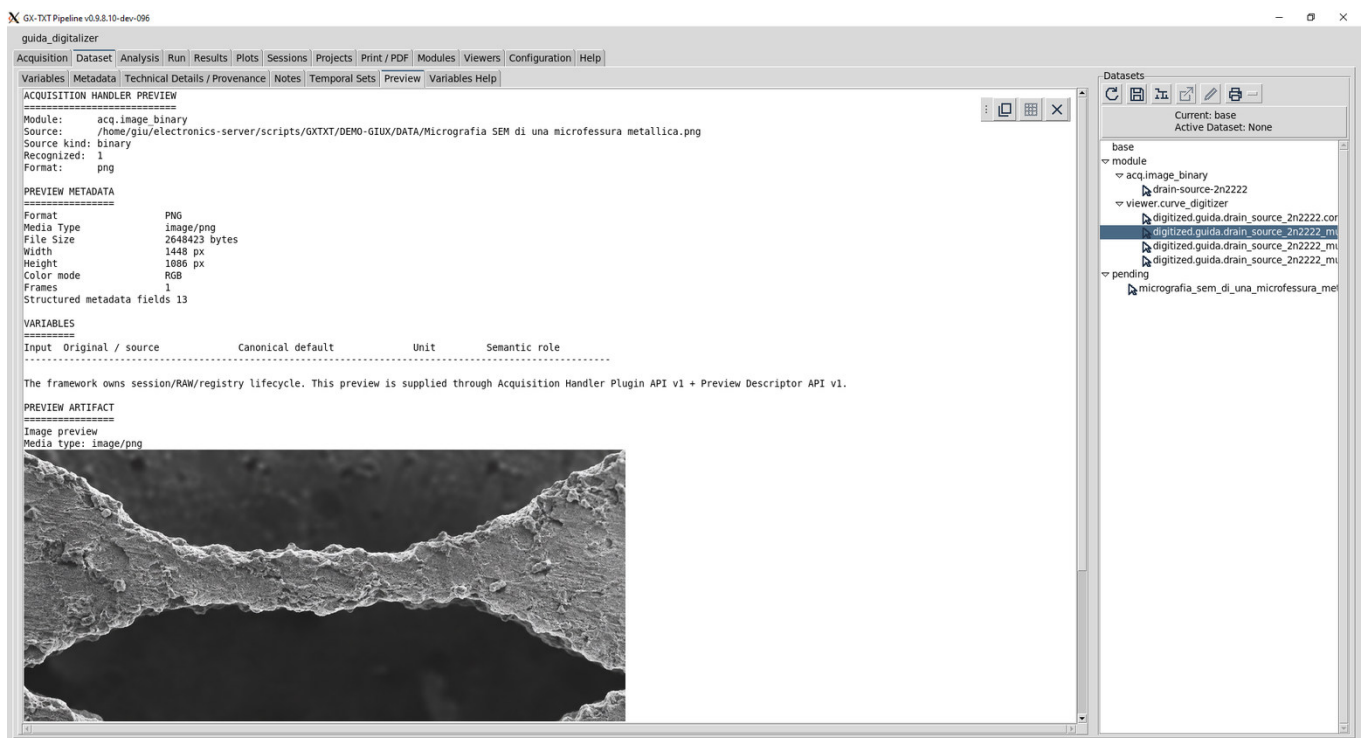


Figure 23. The SEM-style raster visible in the normal GX-TXT acquisition/preview workflow.

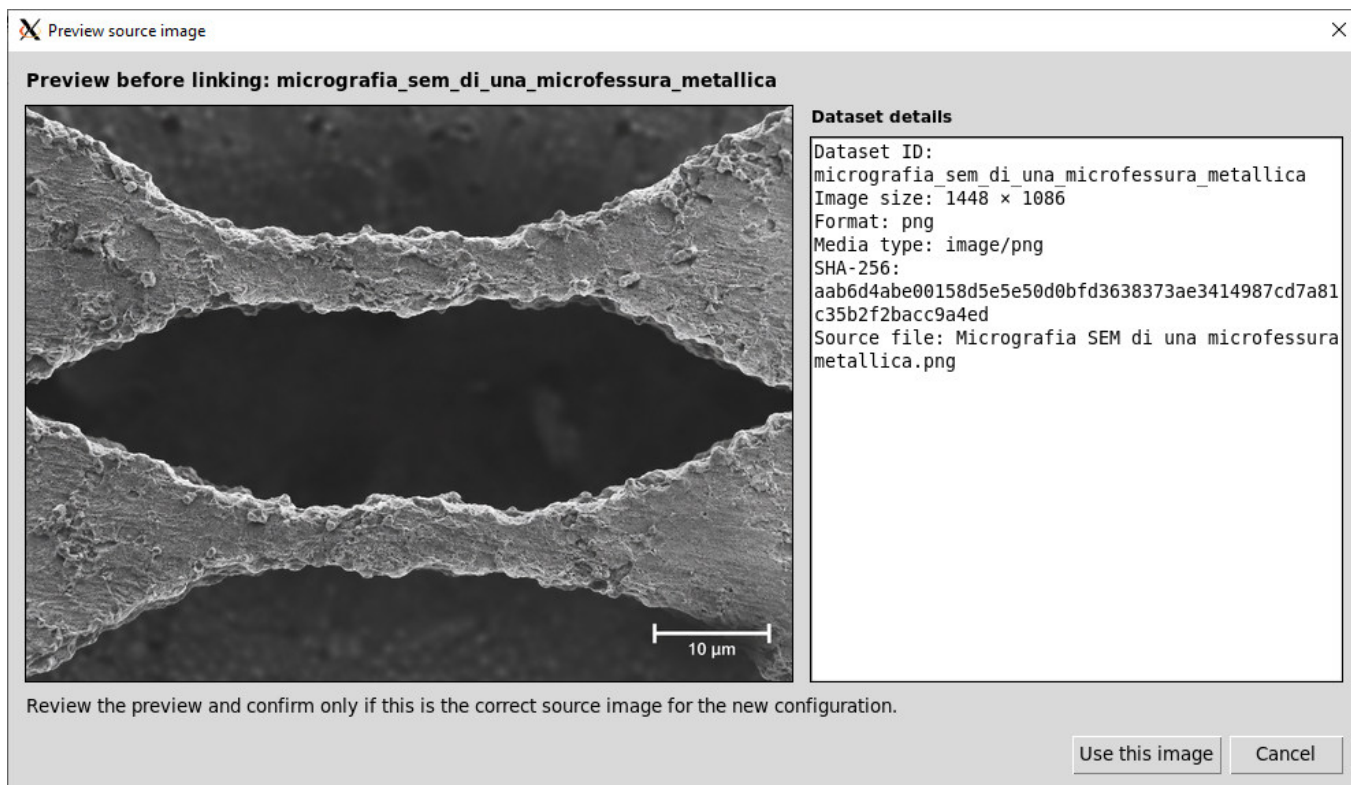


Figure 24. The Curve Digitizer source-preview dialog, including source Dataset ID, raster dimensions, media type and SHA-256.

Create a separate Digitizer configuration for this SEM source and confirm that the image appears unmodified in the viewer.

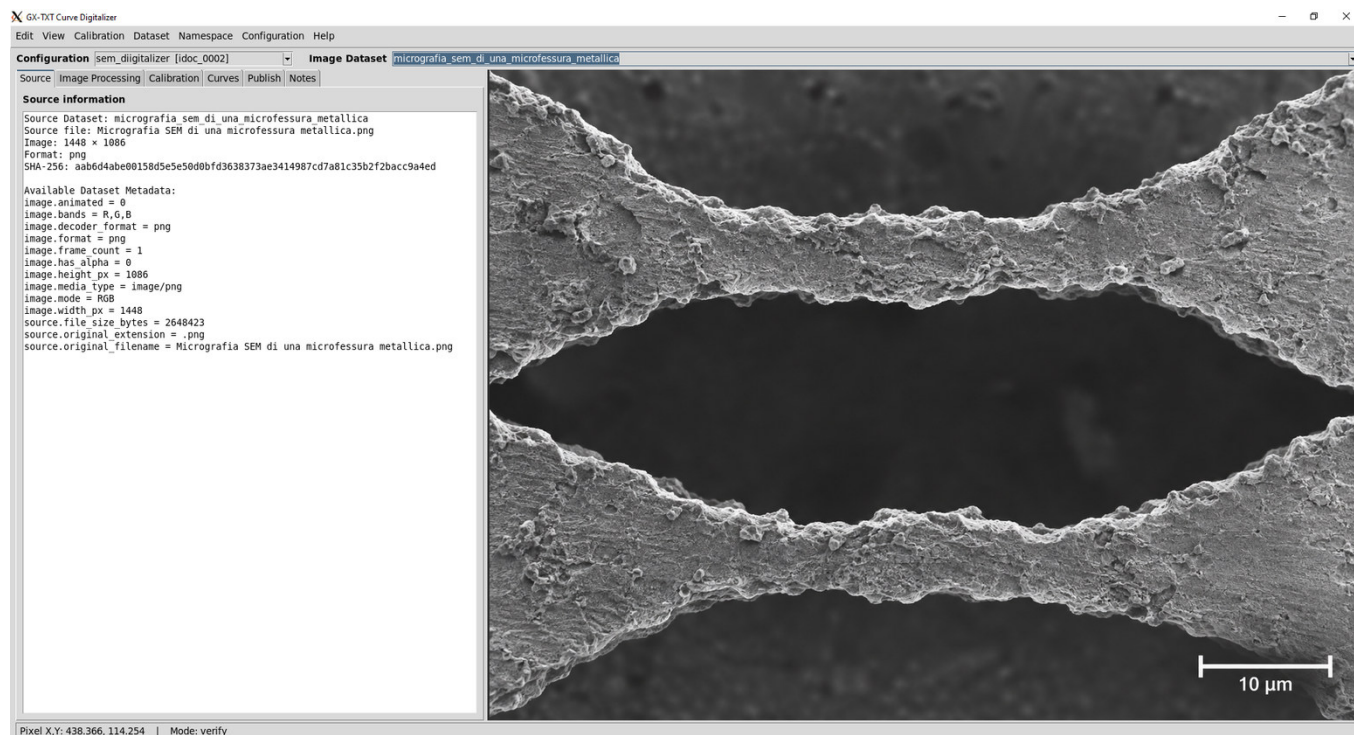


Figure 25. SEM micrograph loaded into its own Curve Digitizer configuration before physical calibration.

2. Auto-calibrate the entire photograph in pixels

Open **Calibration** and choose **Auto-calibrate full image**. Unlike a printed graph, the useful geometric region is the complete raster, so manually placing all four external image corners is unnecessary. This defines a full-image pixel coordinate system. The Y orientation can be selected; the tutorial uses **Upward (Cartesian)**.

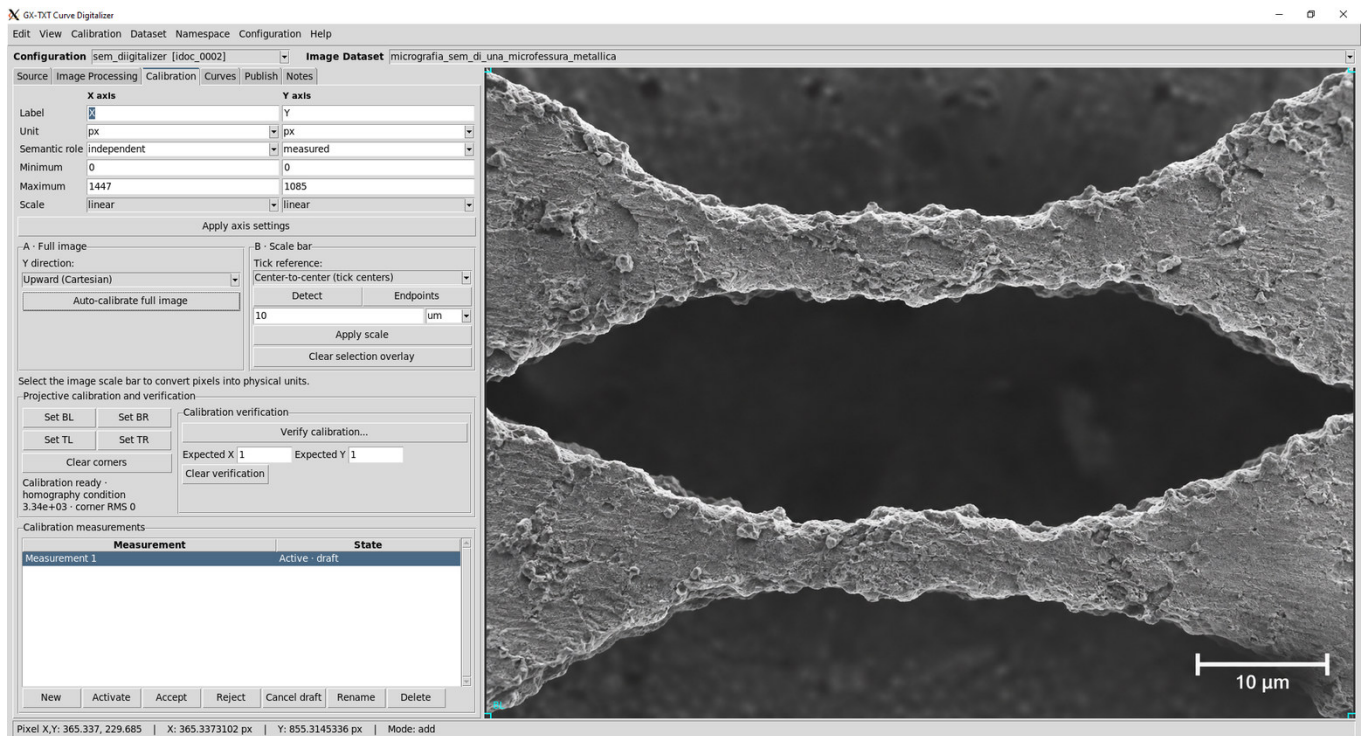


Figure 26. The full raster mapped automatically into pixel coordinates before the physical scale is applied.

3. Detect the 10 µm scale bar

Choose **Detect bar (click image)** and click close to the printed bar. The default reference for a two-tick SEM scale bar is **center-to-center**; if a microscope uses a different convention, choose the appropriate tick reference instead. The selected segment appears as an overlay for visual verification.

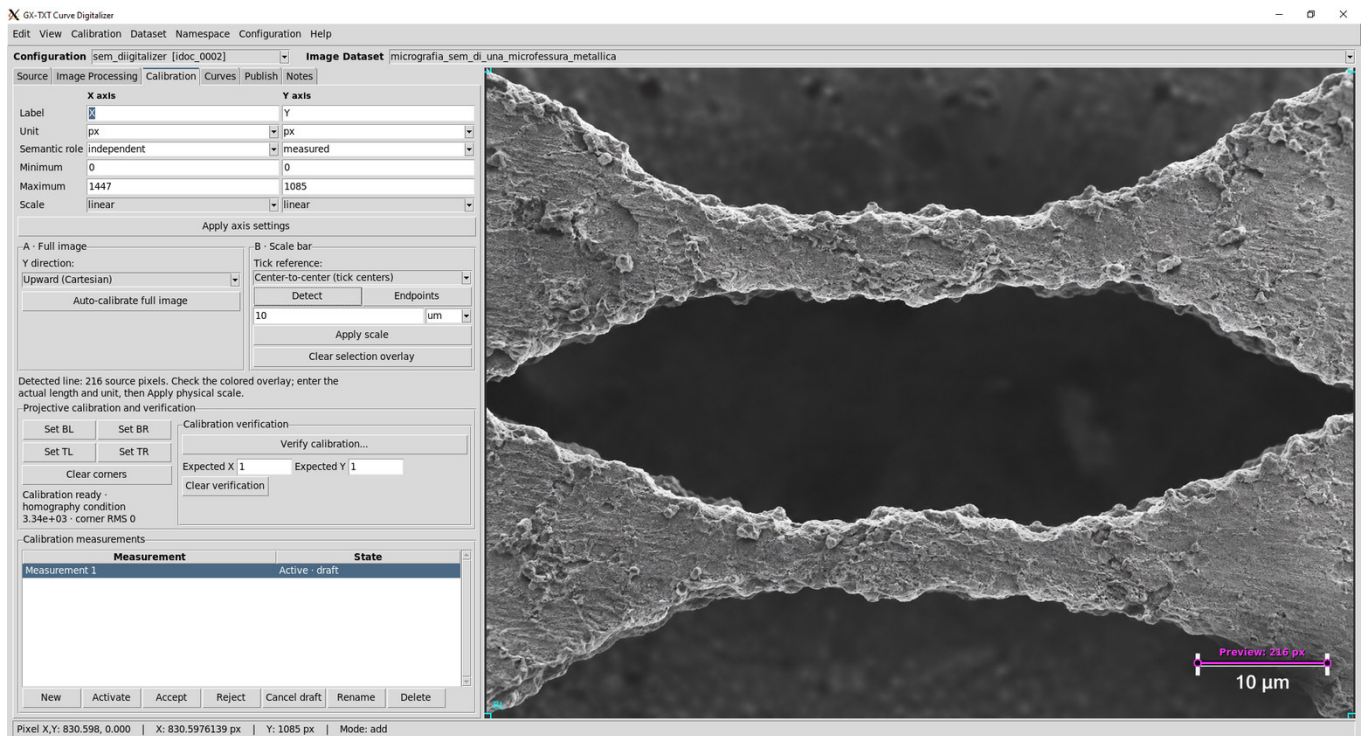


Figure 27. Automatic scale-bar detection. Check the overlay position and endpoints before committing a physical distance.

If the detector is ambiguous, use **Two endpoints** and place the reference manually. Enter the declared physical length as **10** and the unit as **µm**. Only the microscope metadata or an independently trusted bar can establish absolute scale accuracy; this synthetic teaching image is illustrative.

4. Convert the raster to physical coordinates

Click **Apply physical scale**. The scientific axes now show a physical unit while raster-point storage remains in the original image coordinate system. The colored overlay confirms the segment used for the conversion. Clearing only the visual selection should not silently remove the applied calibration.

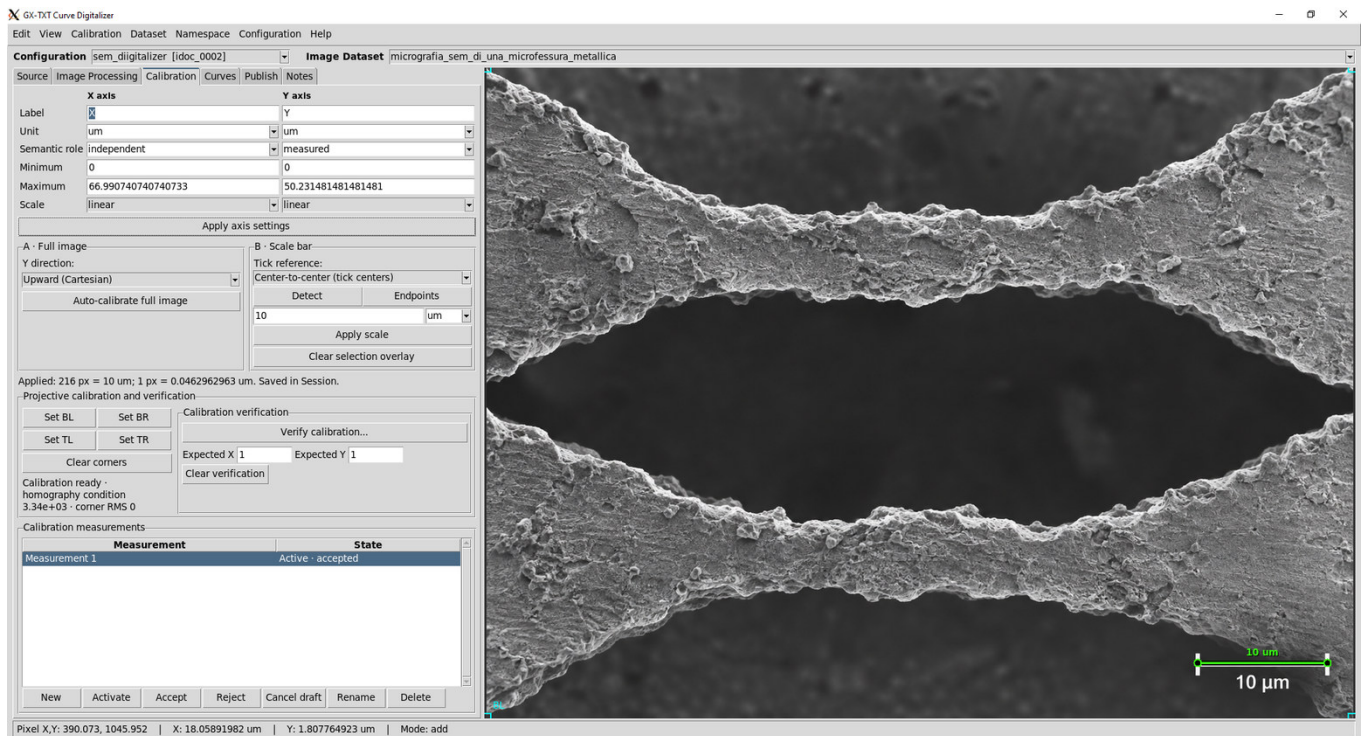


Figure 28. The selected scale bar has been converted into micrometric axis scaling, with the segment visibly marked.

5. Verify a known micrometric position

The verification tool places a temporary red X at a clicked pixel, reports the calculated physical position and supports independent Expected X/Y values. This is the place to detect sign, origin or scale misunderstandings before tracing profiles. Verification never edits curve points.

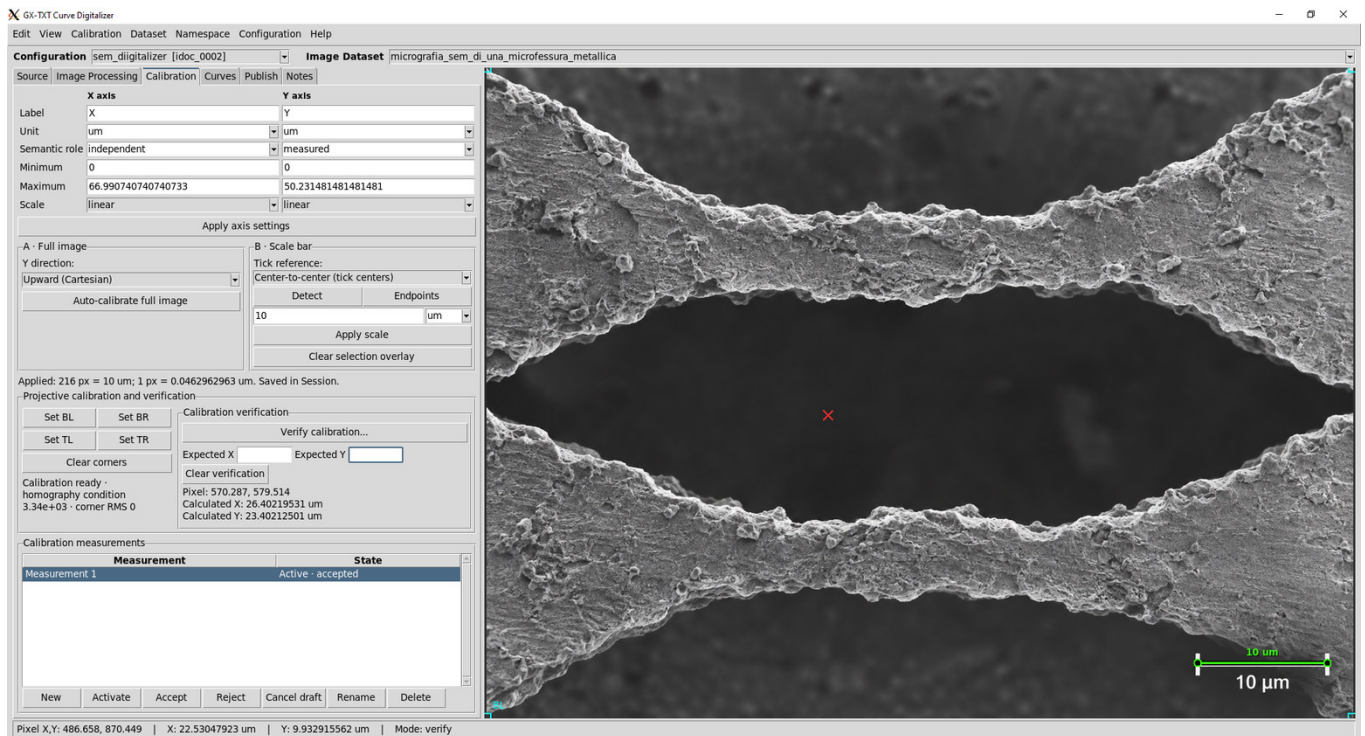


Figure 29. Scale verification on the SEM-style photograph, showing the temporary reference marker and calibrated coordinates.

6. Compare original and processed images

Open **Image Processing**. Only appearance-preserving operations are supported: contrast, gamma, color-space/channel selection, local contrast, denoising, sharpening and thresholding. Operations such as crop, rotation or perspective correction are deliberately outside the Digitizer. Choose **Compare original and processed images** to open the two independently navigable viewports.

For this tutorial the image processing screenshot uses the following active controls: **Grayscale**, **Auto levels**, **Gamma 0.85**, **CLAHE grid 8×8 / bins 128 / clip 3** and **Sharpen sigma 3.2**. The other filters remain disabled. Sharpening is intentionally strong enough here to make the visual difference obvious; do not assume the processed edge position is a more accurate measurement.

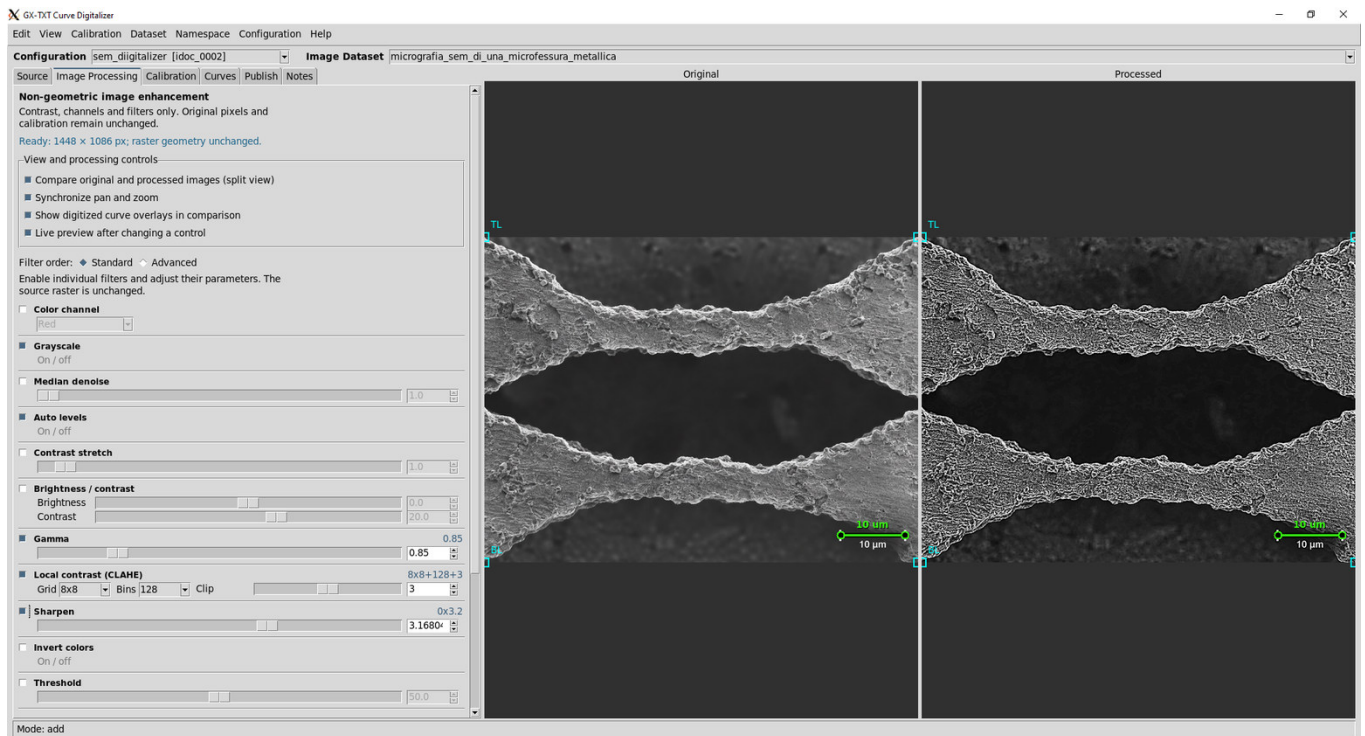


Figure 30. Standard processing controls with synchronized original/processed split comparison. The processed pane emphasizes microtexture while the original remains available for judgement.

The central divider can be dragged; pan and zoom are independent unless **Synchronize pan and zoom** is enabled. The two panes share the same raster geometry, but neither allows editing curve points. Point editing is available only on the Curves tab.

7. Inspect the actual processing order in Advanced mode

Switch from **Standard** to **Advanced** to inspect the exact operation sequence. Order matters because filtering is generally not commutative: applying sharpening before denoising can emphasize unwanted noise; thresholding before tonal adjustment can discard information irreversibly.

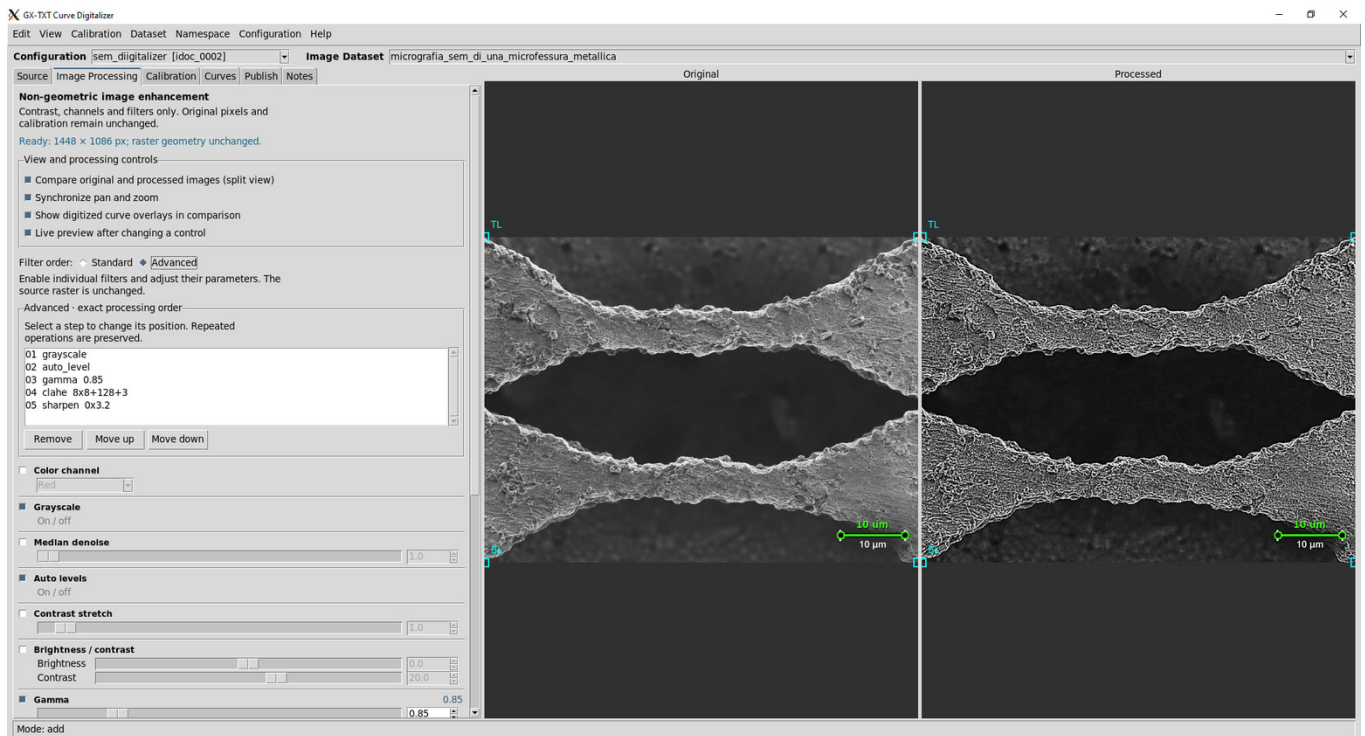


Figure 31. Advanced mode showing the ordered processing pipeline: grayscale, auto levels, gamma, CLAHE and sharpen.

The sequence and filter parameters are recorded for reproducibility. Processing changes image appearance but does not change the raster dimensions, original source checksum or calibration transformation.

8. Publish an independent processed Image Dataset

Scroll to the Image Processing publication controls, provide the output Dataset ID **sem_example_processed** and the descriptive label **Processed SEM example image**, and choose **Publish image.raster Dataset...** Curve publication and processed-image publication are separate choices.

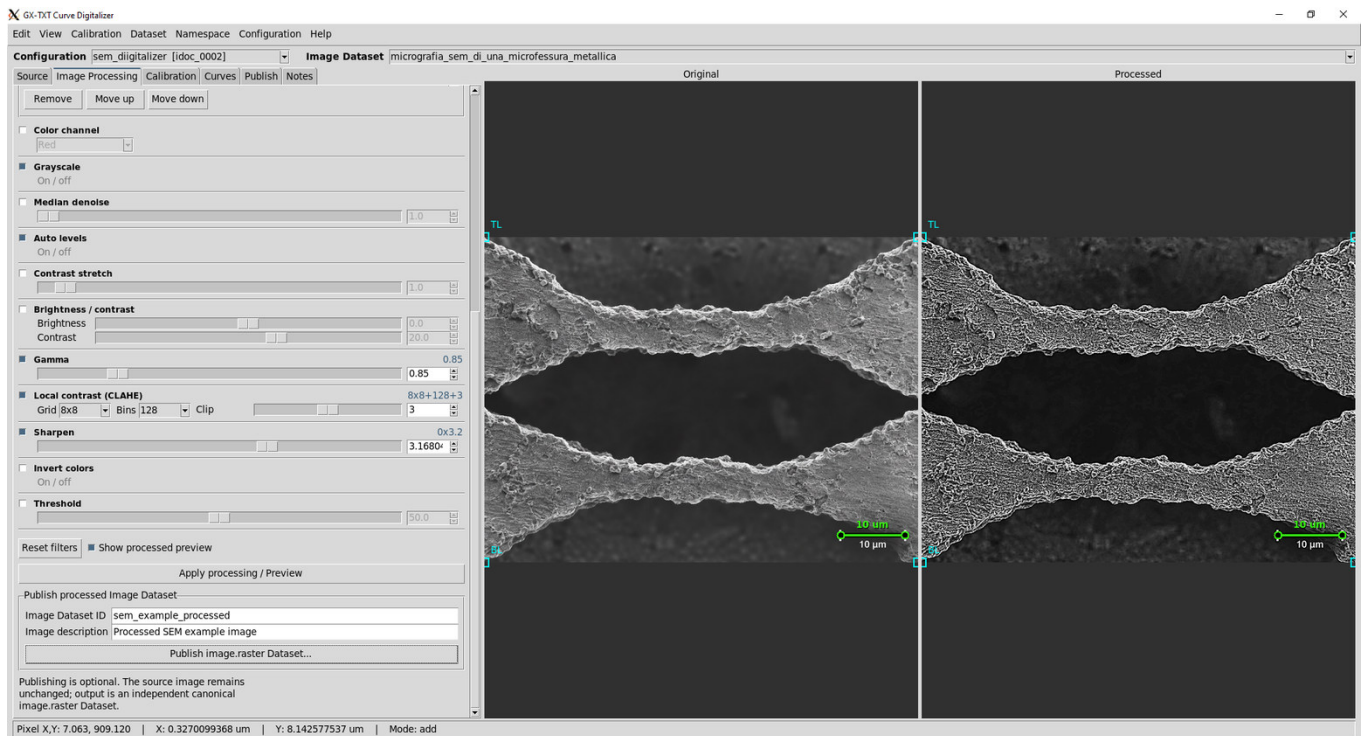


Figure 32. The processed-image publication form shown alongside the original and enhanced raster views.

This action creates a canonical **image.raster** Dataset using GX-TXT file-publication APIs. Its structured metadata records source Dataset identity, original SHA-256, ImageMagick engine/version, exact ordered steps and parameters, geometry-preserved status and derived-image checksum. Other viewers should use the public Image Processing API to read this information rather than hardcode metadata keys.

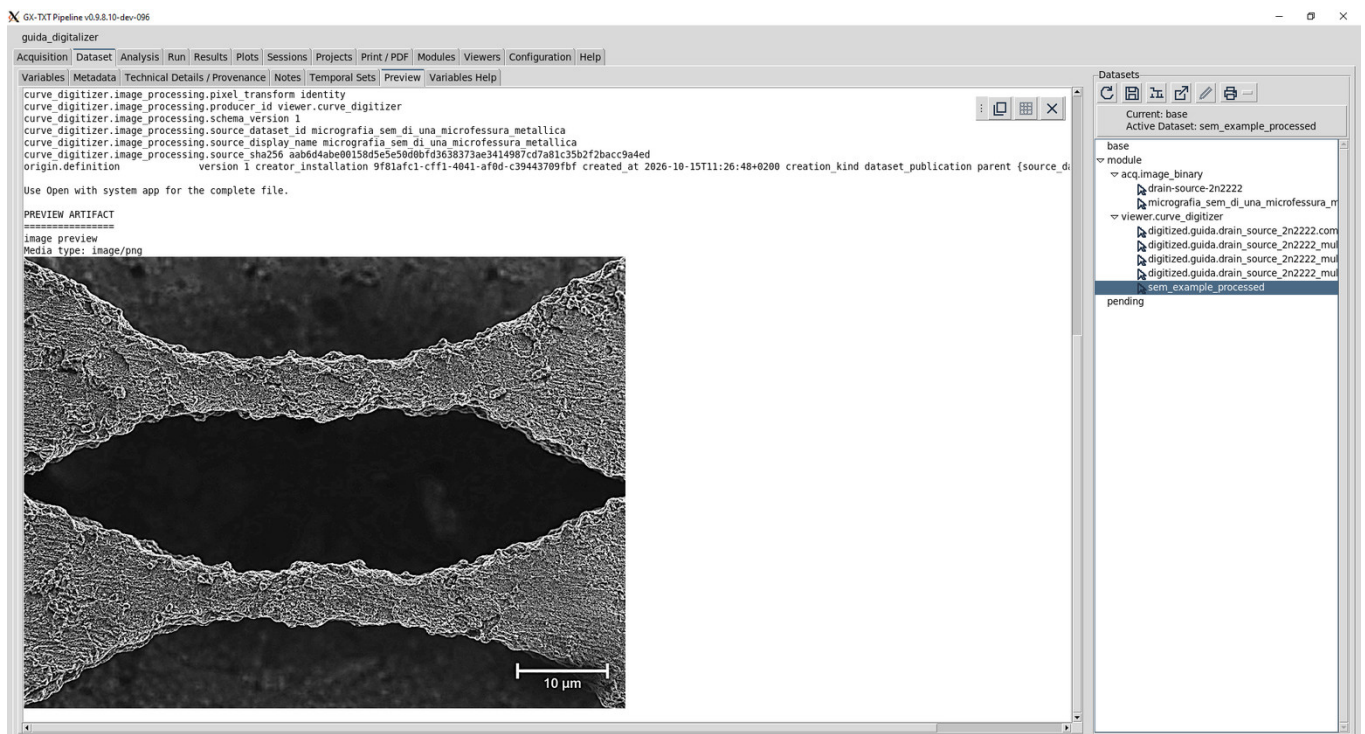


Figure 33. The processed image opened independently from Curve Digitizer through the GX-TXT Dataset browser.

9. Use the processed raster for profile digitization

Create a new Digitizer configuration and select the published enhanced raster as its image source. You may now trace either edge of the microfissure, keeping the original and parent-child source relationship in the Dataset lineage. When finished, publish the profile as a normal curve Dataset with its calibration and pixel coordinates.

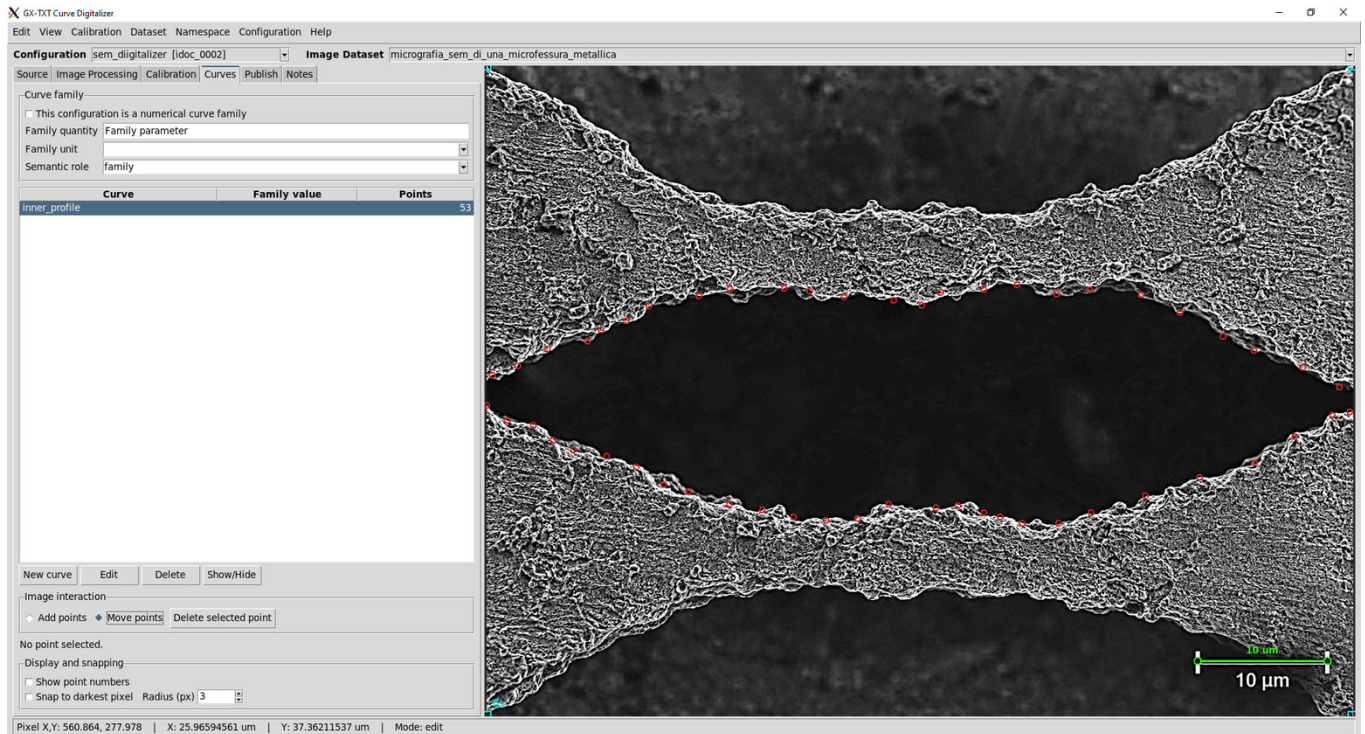


Figure 34. A profile digitized on the processed image after its independent publication as image.raster.

End-to-end SEM result

The second workflow has demonstrated image import, full-image raster calibration, physical scale-bar conversion, independent verification, non-geometric image enhancement, split comparison, ordered processing, publication of a derived image and a profile traced on that published image.

PART III — Detailed viewer reference

The next sections document the implementation actually present in version 1.2.17. First, the configuration screens supplied for this manual are explained in their intended context. After that, the complete installed **MANUAL.md** follows in full rather than a shortened checklist. It covers GUI behavior, persistence, publication contracts, metadata, error recovery and all three public API families.

GX-TXT central module configuration

From GX-TXT Configuration open the Modules configuration center, select **Curve Digitalizer** and edit the user defaults or Session overrides. The module declares typed configuration fields through **CONFIGURATION_API_v1**; there is no separate private settings database. Changes are reconciled by the running viewer.

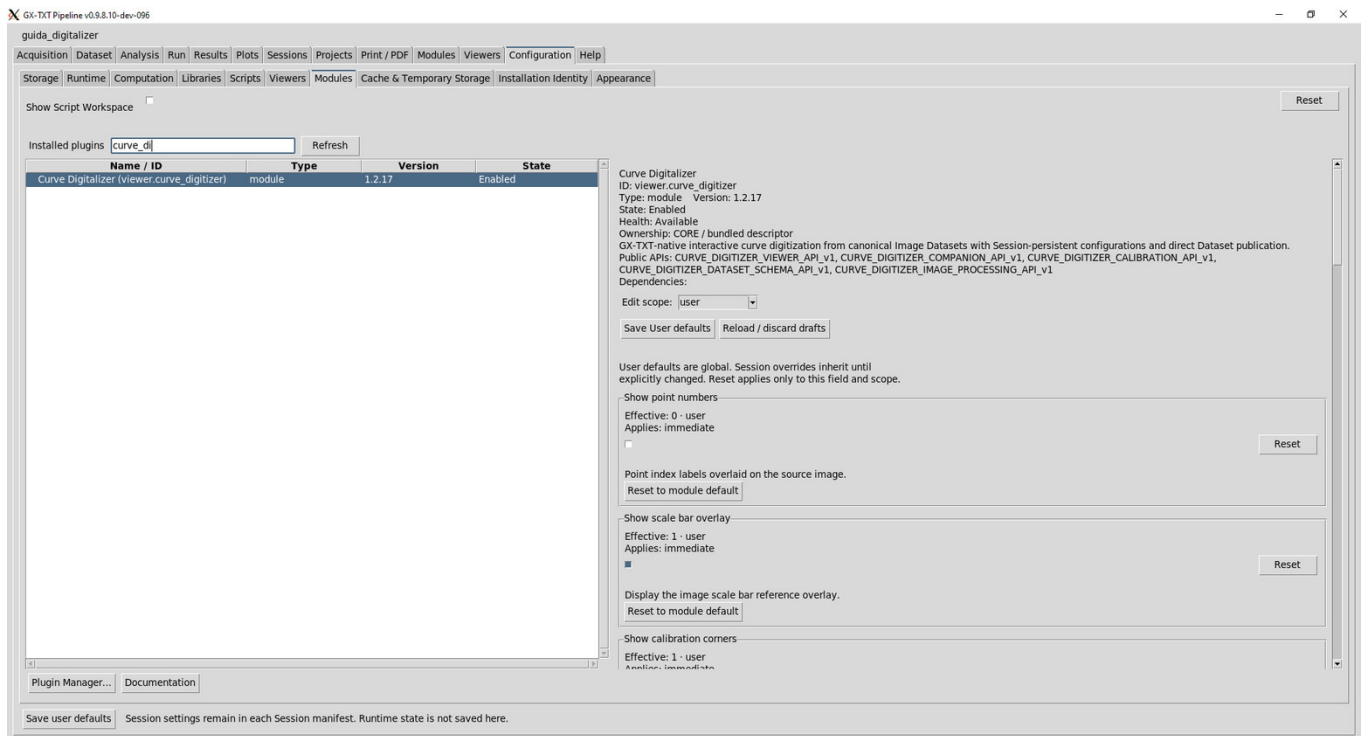


Figure 35. GX-TXT central configuration page for installed modules, including Curve Digitizer configuration controls.

Viewer settings: Display

Display settings control the visibility of point numbers, scale-bar overlays and projective-calibration corner markers. They affect screen presentation, not the underlying scientific image or digitized point coordinates.

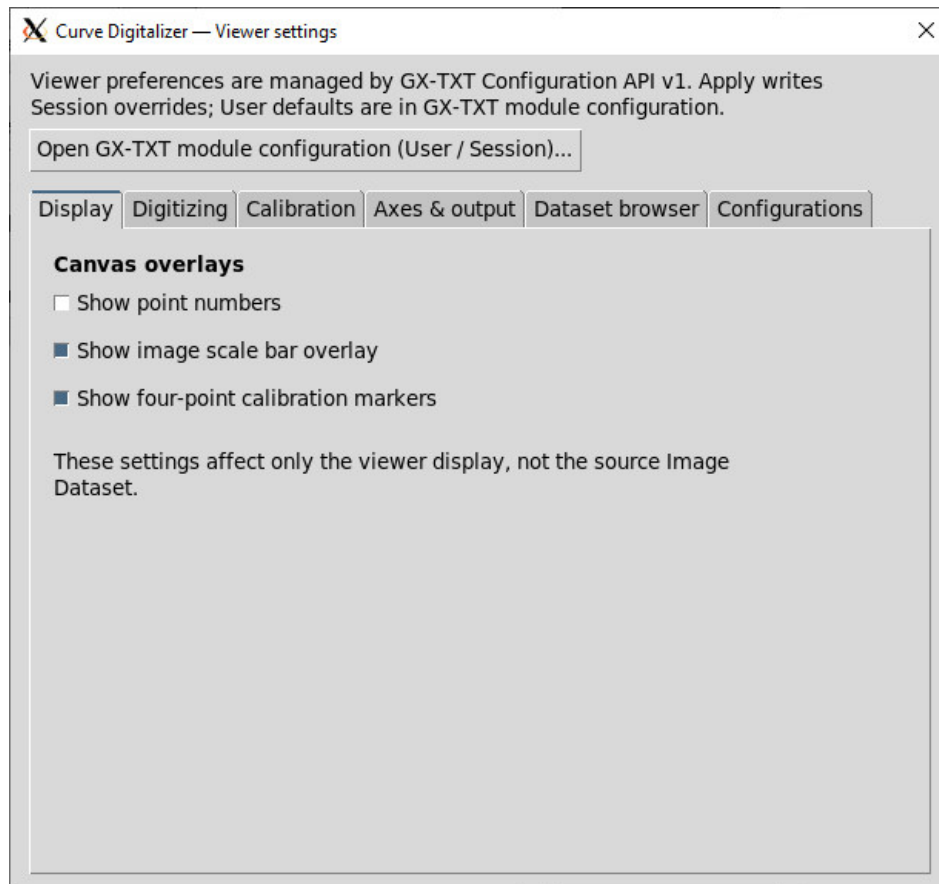


Figure 36. The Display page of the viewer settings dialog.

Viewer settings: Digitizing

Digitizing settings include optional snapping to the darkest pixel and the radius used for its neighborhood search. Snapping can help on raster traces, but it can also select grid lines or noise; use it only when the source justifies the choice.

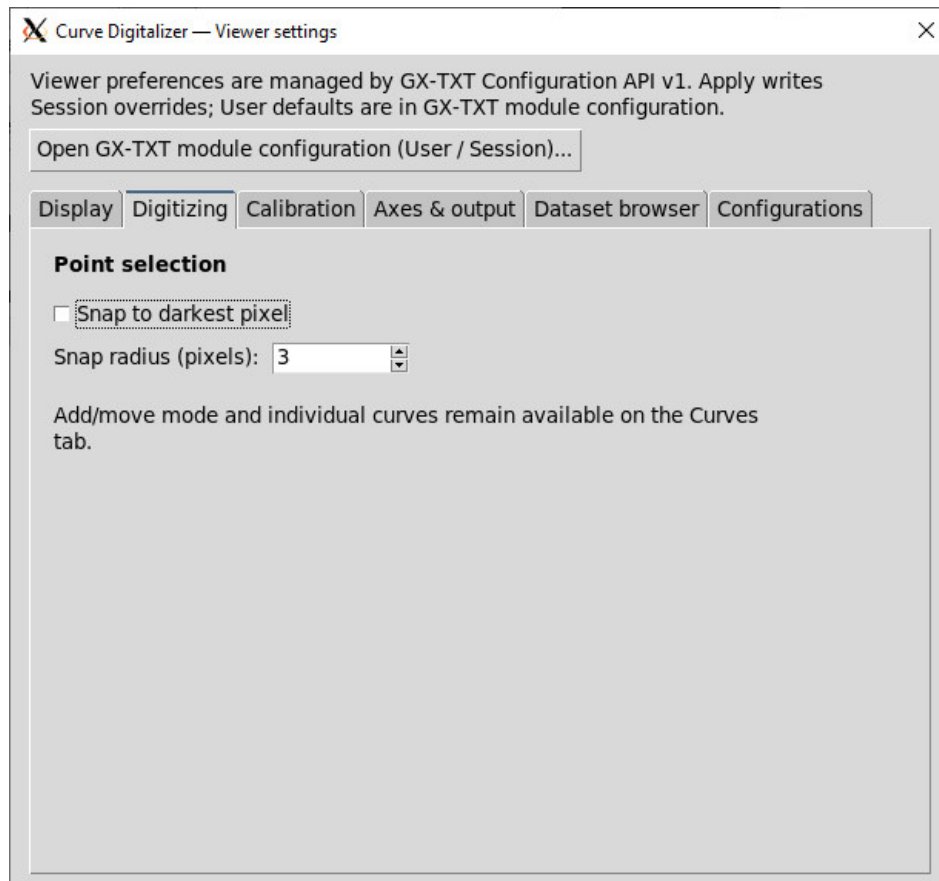


Figure 37. The Digitizing page, including the point snapping control and radius.

Viewer settings: Calibration

Calibration settings provide scale-bar tick reference behavior, verification-marker appearance, full-image Y orientation and initial physical scale preferences. These are defaults for future interactions: modifying settings does not secretly replace an already accepted calibration.

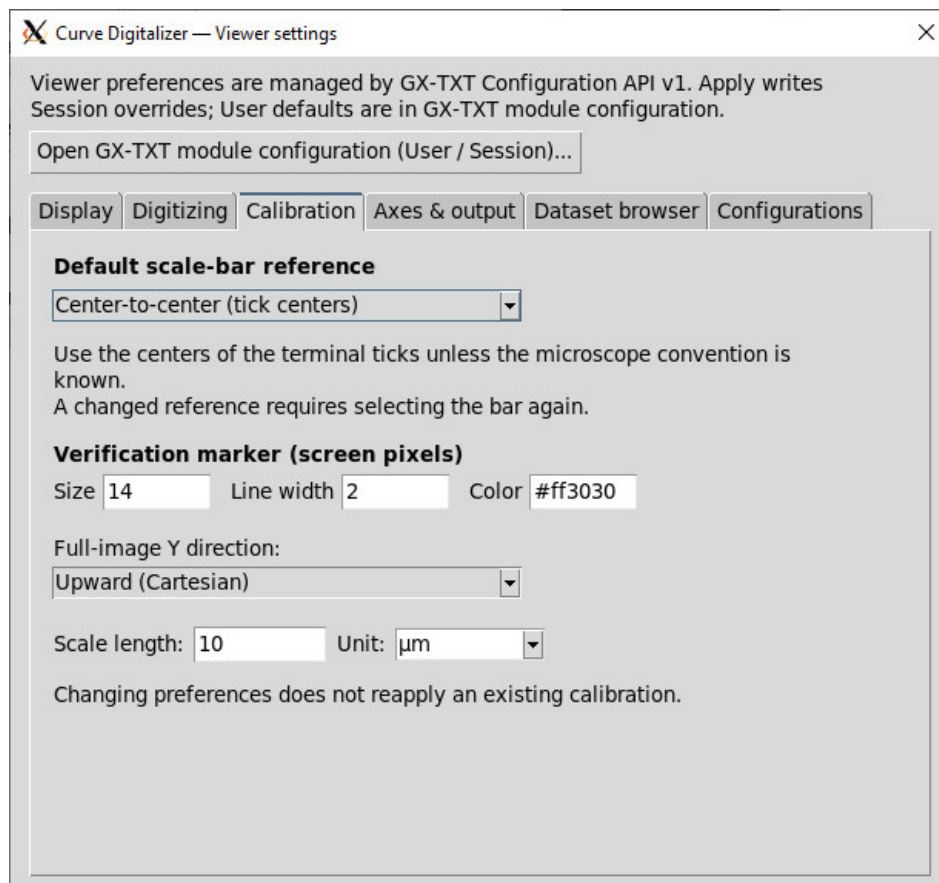


Figure 38. The Calibration settings page, with center-to-center scale reference and verification marker settings.

Viewer settings: Axes and output

The output page displays the axis configuration and offers controls for publication identity, Separate/Combined defaults and Dataset ID naming templates. The template may include placeholders such as **[name]**, **[calibration]** and **[configuration]**, with the final identity validated by GX-TXT before publication.

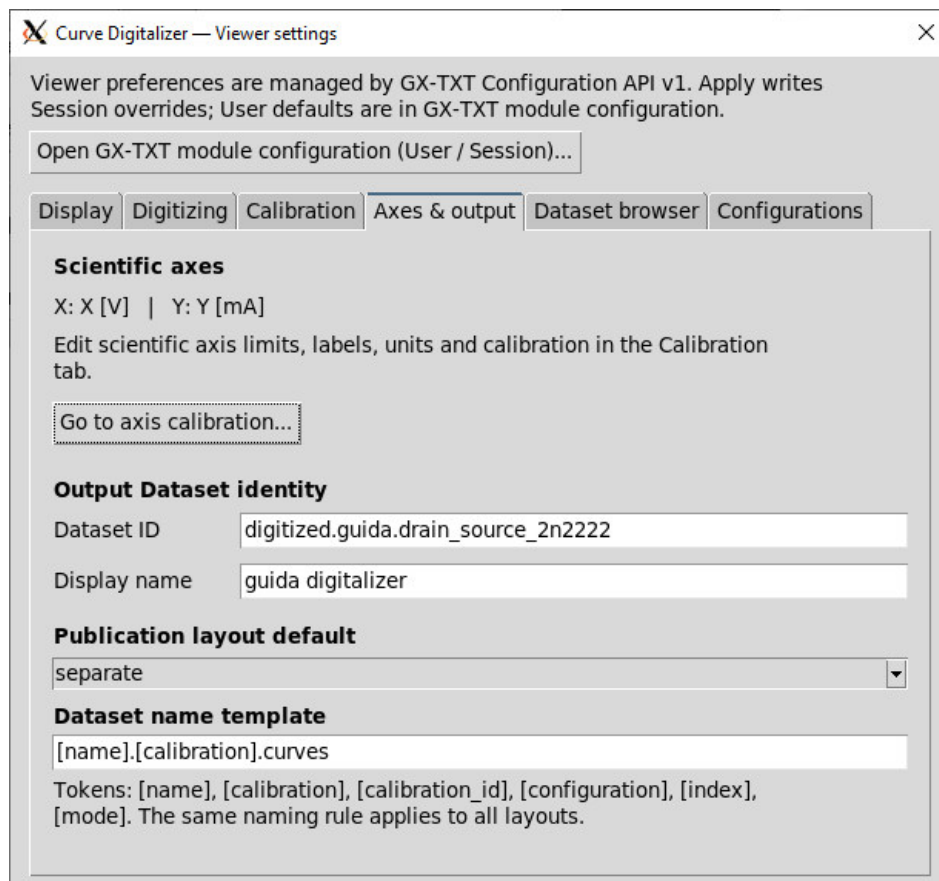


Figure 39. Axes and output settings, including the Dataset naming template and publication defaults.

Viewer settings: Dataset browser

Automatic catalog monitoring and namespace filtering belong to Dataset browser settings. Choose the namespace using the Namespace menu; this filter determines which compatible image Datasets are shown without automatically replacing the current source. Missing or stale source images are protected against silent reassociation.

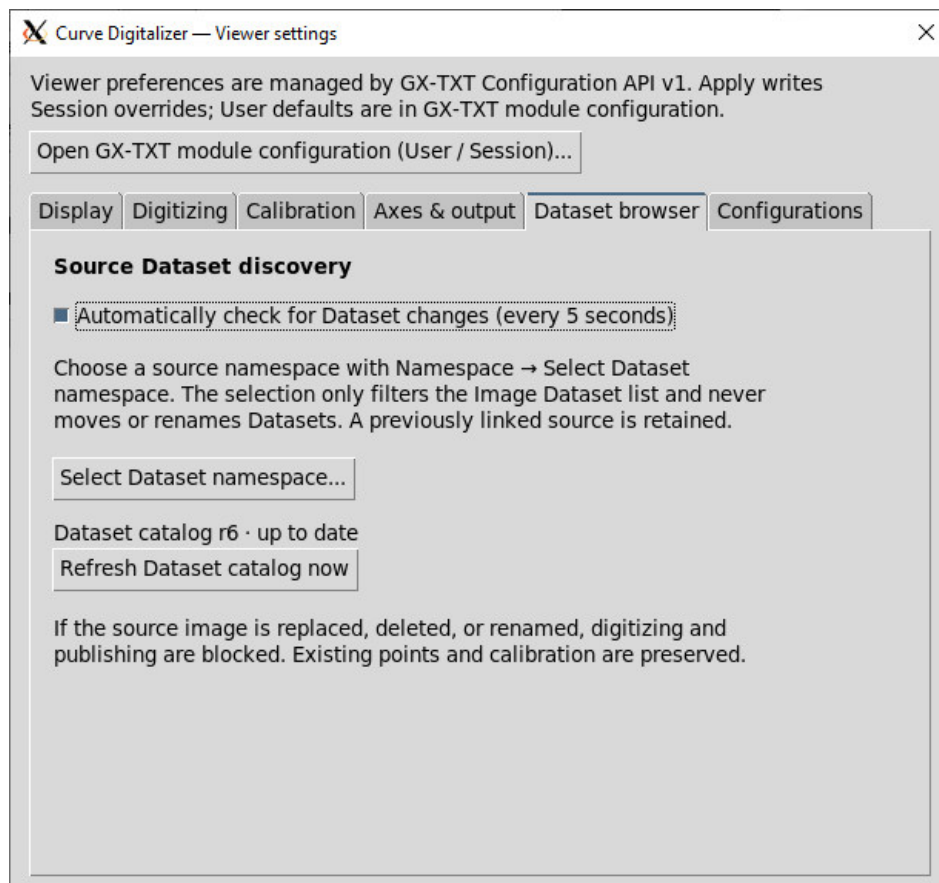


Figure 40. Dataset browser settings, including automatic refresh and the namespace selection action.

Viewer settings: Configurations

Configurations are persistent scientific work objects stored inside the current Session. The settings dialog presents their names and identities. Use the menu actions to create, rename, duplicate or delete work objects, and use the selector in the main header to switch between them.

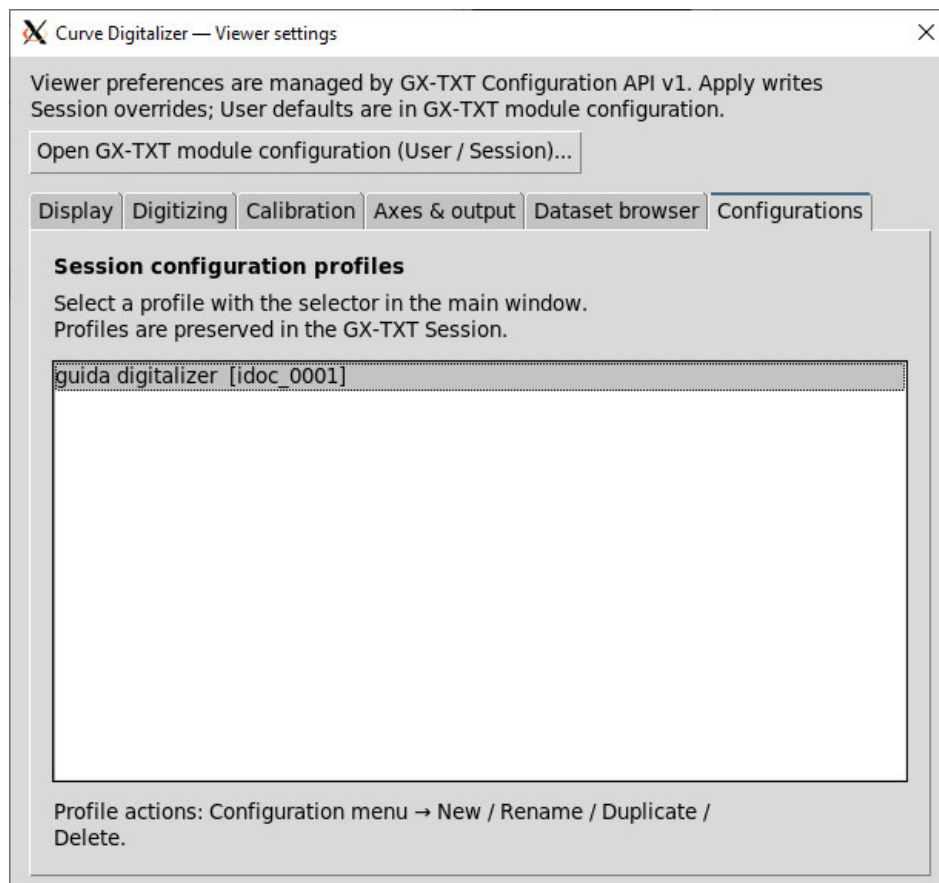


Figure 41. The Configurations page listing the Session-owned digitization configurations.

APPENDIX A — Full installed viewer manual

The pages below are a typographic rendering of **MANUAL.md** taken directly from the installed Curve Digitalizer **1.2.17** plugin package. This appendix is intentionally complete: no feature list, procedural text, metadata contract, technical limitation or API example is removed for brevity. Its chapter numbering follows the source material.

Curve Digitalizer Manual

1. Purpose

Curve Digitalizer is an interactive GX-TXT Viewer for recovering numerical data from curves drawn in raster images. Typical sources include scanned electronics datasheets, historical plots, instrument screenshots, and published scientific figures.

The module is data-centric. The image is a GX-TXT Dataset, the digitization work is authoritative Session state, and the scientific result is a normal GX-TXT Dataset that can be consumed without reopening the Curve Digitalizer UI.

2. Architecture and normal workflow

```
Image file
-> GX-TXT acquisition/import
-> Canonical Image Dataset
-> Curve Digitalizer Viewer
-> named Digitizer Configuration
-> calibration + curves + points + metadata
-> Session persistence
-> Publish Dataset
-> CSV scientific artifact + Canonical Dataset
```

The standalone operations Open Image, Save Project, Load Project, and manual Export CSV are intentionally not part of normal GX-TXT operation.

3. Supported source Image Datasets

Curve Digitalizer discovers compatible image Datasets in the current Session through the public Dataset registry and resolution APIs. A Dataset is considered compatible when it exposes the canonical `image.raster` contract or an image media type.

The `acq.image_binary` importer normally imports PNG, JPEG, BMP, and TIFF and preserves the original canonical resource. Curve Digitalizer reads that canonical resource through the public Dataset API. It does not inspect or construct private Session paths.

The desktop adapter uses Pillow to decode the resolved image resource.

4. Opening the viewer

Open an existing GX-TXT Session, import an image through the normal image acquisition workflow, then open:

```
Viewers -> Curve Digitalizer
```

The Viewer Provider is discovered headlessly. Python/Tk/NumPy/Pillow are checked only when the interactive viewer is opened.

If no Session is open, the viewer reports an error instead of creating private state outside GX-TXT.

4.1 Dataset workspace selector and automatic updates

Select **Namespace** → **Select Dataset namespace...** to open the hierarchical Dataset namespace picker. It lists GX-TXT Dataset namespaces. These are the framework's organizational groups for canonical Datasets; they are not separate directories, and are distinct from Session Workspaces. Choose **All namespaces** to see every compatible image in the Session. Choosing an organizational namespace shows images stored in that namespace and its descendants. The **Image Dataset** selector in the compact header lists matching Image Datasets.

The viewer checks the **Dataset Catalog revision** every five seconds when the **Check Dataset changes automatically** preference is enabled. Checks use the public Dataset Catalog API and only read a complete

catalog snapshot after the revision changes. You can also choose **Dataset → Refresh Dataset catalog**. The settings and interval are explained in **Configuration → Viewer settings → Dataset browser**.

New, removed, renamed or relocated Datasets update the source choices. Moving between workspace filters **does not** change a saved configuration's source Dataset ID, even if the source is hidden by the current filter. If an image's contents change, the viewer verifies its actual SHA-256 against the digest recorded at original selection. A mismatch marks **STALE SOURCE** and disables new scientific edits and publication. A deleted/renamed Dataset marks **SOURCE MISSING**. Existing points, calibrations and metadata remain preserved, and must not be silently reassociated with another source. Create a new configuration for a new source rather than assuming the same image coordinates still apply. A Session change similarly invalidates an already opened viewer.

5. Digitizer Configurations

A Digitizer Configuration is the authoritative scientific work object for a set of curves and a group of calibration measurements associated with one source image. It is stored in the current Session as a GX-TXT Interactive Document.

Each configuration has:

- a stable generated configuration ID;
- a user-visible name;
- one source Dataset ID;
- source resource identity and SHA-256;
- X/Y axis definitions;
- projective calibration control points;
- curves and stable curve IDs;
- ordered digitized points;
- optional family definition;
- display/snapping state;
- publication settings;
- optional notes;
- a persistent interactive document format.

Use the configuration selector at the top of the viewer to switch between configurations.

5.1 New

Select **New** and enter a name. Then select one of the compatible source Image Datasets from the current Session.

A source is associated with a configuration only after explicit selection.

5.2 Rename

Select **Rename** to change the display name. The configuration ID remains unchanged.

5.3 Duplicate

Select **Duplicate** to create an independent configuration containing a copy of the current state. The duplicate receives a new stable configuration ID.

5.4 Delete

Select **Delete** and confirm. Only that configuration's Session-persisted state is removed. Already published Datasets are not silently deleted.

5.5 Multiple configurations per image

The same Image Dataset can back several independent configurations. For example, one scanned datasheet page may contain plate characteristics, transfer characteristics, and grid-current curves. Each configuration can use a different calibration and publish to a different Dataset.

Changing a calibrated configuration to another source Dataset is deliberately blocked. Create a new configuration or duplicate/reset explicitly instead; calibration is pixel-coordinate-dependent.

6. Source Dataset information and metadata

When a source Dataset is selected, Curve Digitalizer reads Dataset Metadata through the public Dataset Metadata API.

The Source information panel displays only information actually available, such as:

- Dataset ID;
- canonical format/media type;
- image dimensions if supplied;
- source filename or document information;
- page information;
- manufacturer or datasheet metadata;
- URL/date/scan information;
- other namespaced metadata.

The module does not require specific optional metadata keys and works when optional metadata is absent.

7. Source integrity and stale-source protection

A calibration is meaningful only for the exact pixels from which it was created. The configuration therefore stores the source Dataset identity and source SHA-256.

Whenever the configuration is opened, Curve Digitalizer resolves the current source through the public Dataset API and hashes the bytes actually returned.

Possible states are:

- **Source Dataset verified by SHA-256:** editing and publication are allowed.
- **STALE SOURCE:** the Dataset ID still exists but the current image bytes differ. Existing calibration/curves remain preserved, but calibrated work and publication are disabled until the situation is resolved explicitly.
- **SOURCE MISSING:** the Dataset no longer exists or cannot be resolved. The digitization configuration remains authoritative Session state and is not deleted.

The module never silently applies old pixel calibration to changed source bytes.

8. Axis calibration

Curve Digitalizer preserves the original four-point projective/homographic calibration model.

8.1 Axis definitions

For each axis specify:

- display label;
- unit;
- semantic role;
- minimum value;
- maximum value;
- scale mode: linear or log.

Units are offered from the GX-TXT Unit Registry, but the field remains editable. A textual unit not recognized by the registry is preserved explicitly rather than discarded.

Semantic roles are offered from the current Semantic Registry and are not inferred from axis labels.

8.2 Linear and logarithmic combinations

All four combinations are supported:

```
linear X / linear Y
linear X / log Y
log X / linear Y
log X / log Y
```

Logarithmic axes require positive scientific limits.

8.3 Automatic full-image calibration for photographs

The **Calibration** tab contains a **Full image (photographs)** section. After selecting a source Image Dataset, click **Auto-calibrate full image** to calibrate the entire decoded raster automatically, without manual corner clicks. It uses the actual Pillow image dimensions, not the displayed/zoomed canvas dimensions or optional metadata.

- Four calibration points are set to the pixel **centres** at the actual image corners: bottom-left (0, H-1), bottom-right (W-1, H-1), top-left (0, 0), and top-right (W-1, 0).
- The X axis becomes linear, 0 through W-1, with unit px.

- The Y axis becomes linear, with unit px. Choose **Upward (Cartesian)** for $Y=0$ at the bottom and $Y=H-1$ at the top; choose **Downward (image pixels)** for $Y=0$ at the top and $Y=H-1$ at the bottom.
- Custom X/Y labels and semantic roles remain unchanged. Existing digitized points are **recalculated**, not removed. Replacing a current calibration or customized axis limits requires confirmation and can be undone or redone.
- Configuration state uses the same Session Interactive Document, source SHA-256 protections, projective transform, and publication path as manual calibration. There is no independent calibration/persistence mechanism.
- If the source is unavailable or stale, automatic calibration is blocked. Images narrower than two pixels or shorter than two pixels cannot be calibrated.

The **Four-point projective calibration (manual)** section remains available below the image scaling controls for graphs, image crops, skewed plots, or arbitrary quadrilaterals. Reapplying manual calibration can override automatic corner locations. The automatic pixel bounds alone do not infer physical scale; use **Scale using an image scale bar** for calibrated SEM micrographs and photographs.

8.4 Scale using an image scale bar (photographs and SEM)

This feature is a second, **explicit** action after **Auto-calibrate full image**. It is not a perspective correction and does not assume a physical scale from image pixels alone.

1. Choose the canonical source image and press **Auto-calibrate full image**; select the desired Y direction.
2. Select the **Tick reference**. Default: **Center-to-center (tick centers)**. Inner/outer edges are alternatives only when the microscope's scale-bar convention is known. Click **Detect**, then click near the middle of a visually distinct scale bar on the source image (typically below the SEM micrograph). The algorithm compares the bar with nearby background and searches for a long horizontal light or dark line. It does not require the bar to be white, but does require sufficient contrast. Only bars visible in the source pixels are eligible; canvas zoom does not affect measurement.
3. Inspect the **magenta overlay**, including the two outlined endpoints. The candidate pixel length is displayed in the Calibration tab. The candidate is not applied automatically.
4. If the detection is ambiguous, incomplete or selects the wrong segment, press **Endpoints** and click the physical ends of the bar. This fallback also handles rotated bars. A line at the wrong position must never be accepted just because a detector found it.
5. Enter the actual **Actual length** of the bar and its **physical unit**, such as 10 and μm . Press **Apply scale**. The viewer scales X and Y equally; the measured reference is drawn over the image using a **green line with black contrast outline**, endpoint markers and an annotation. The original image is never modified.

The factor is $\text{physical length} / \text{measured Euclidean pixel distance}$ in the selected unit per source pixel. Original pixel centers $(0,0)$ through $(W-1,H-1)$ define the image extent. For example, when a $10\ \mu\text{m}$ bar spans 152 px in a 720×560 image, the factor is $10/152\ \mu\text{m}/\text{px}$ and full-image dimensions in μm become $719 \times 10/152$ by $559 \times 10/152$. The Y direction stays Cartesian-upward or image-downward as chosen for full-image calibration. Scale is assumed isotropic, i.e. original image pixels represent equal physical distances horizontally and vertically.

The scale reference is persisted in the **existing Session configuration** as two source-pixel endpoints, pixel distance, physical length, unit, detection mode (detected or manual) and tick reference (center, inner or outer). Changing the tick reference invalidates a pending preview and requires detecting/selecting the bar again. The previous applied calibration remains unchanged until the new one is explicitly applied. Digitized points retain their original source pixel positions and are recalculated in physical coordinates. Undo/redone works as with other calibration actions. Reopening the Session restores the green reference overlay. A later manual corner

edit, change of physical limits/units, or new full-image autocalibration removes an invalidated previous scale reference; it can be restored with Undo where applicable. Published Datasets keep the normal output Dataset identity and the same GX-TXT Dataset Publication API, with `scale_bar` reported as the calibration type. The detailed reference remains traceable through the linked digitizer configuration document.

Limits: a numerical scale cannot compensate for distorted/non-square pixels, perspective, or microscope calibration errors; it assumes the raster dimensions correspond to the scale bar. Automatic line detection is intentionally conservative: complex photographic backgrounds or low-contrast/fragmented bars can require manual selection. Do not treat a detected line as ground truth before visually checking the overlay.

8.5 Four calibration points

After applying the axis settings, select the calibration points in this order:

```
bottom-left
bottom-right
top-left
top-right
```

The model computes a projective homography from the image quadrilateral to normalized plot coordinates and then maps normalized coordinates through the selected linear/log axis transforms.

The status line reports calibration readiness, conditioning, and corner residual information.

8.6 Calibration verification

Use the verification controls on a known grid intersection or other known coordinate. Curve Digitalizer reports calculated X/Y values and, when expected values are supplied, the residuals.

Verification is diagnostic; it does not change the calibration.

9. Curves

Each curve has a stable internal `curve_id`, a display label, and an ordered list of points.

Use **New curve**, **Edit curve**, and **Delete curve** to manage curves. Visibility can be toggled without deleting data.

Curve labels are not technical identities and may be edited.

10. Numerical curve families

Enable **This configuration is a numerical curve family** only when one real numerical quantity parameterizes the curves.

Example:

```
X: Anode voltage [V]
Y: Anode current [mA]
Family quantity: Grid voltage
Family unit: V
Curve values: -2, -4, -6
```

For family publication:

- every curve must have a numerical family value;
- the configured family unit is the common unit;
- a per-curve unit, when present, must agree with that configured unit;
- the family quantity has an explicit semantic role, normally family.

Do not enable family mode for unrelated curves that happen to be on the same image.

11. Point digitization and editing

11.1 Add points

Select a curve and enter **Add points** mode. Click the curve in the image. Each point stores pixel information and its calibrated scientific coordinates.

Point order is the user digitization order and is preserved during publication.

11.2 Move/edit points

Use the move/edit mode to reposition an existing point. The corresponding scientific coordinates are recomputed from the current calibration.

11.3 Delete points

Select a point and delete it using the viewer control or the Delete key where applicable.

11.4 Undo/redo

Scientific edits push complete configuration snapshots onto the viewer's undo stack. **Undo** and **Redo** restore those states and then persist the resulting authoritative configuration.

11.5 Point numbering

Optional point numbers can be displayed as an overlay. This changes only presentation.

11.6 Darkest-pixel snapping

Optional snapping searches a configurable local pixel radius for the darkest candidate. The original/manual click and the snapped/stored position remain distinguishable in the data model inherited from the original digitalizer.

Snapping is an aid, not an automatic curve recognizer.

12. Zoom, pan, and image transform

The reusable canvas supports:

- mouse-wheel zoom;
- middle/right-button pan;
- fit image to the available canvas.

The interactive transform is presentation state. Calibration and digitized points remain defined against image pixel coordinates, so zoom/pan do not alter scientific values.

13. Session persistence

Authoritative configurations are stored with GX-TXT's Session-scoped Interactive Document API.

Normal persistence is therefore:

```
Save/keep Session state
-> close GX-TXT
-> reopen Session
-> open Curve Digitalizer
-> select the same named configuration
```

The calibration, axis definitions, curves, point order, notes, source identity, publication settings, and other configuration state are restored.

No standalone .json project is required.

Closing the viewer, resetting derived runtime state, or temporarily changing UI selection does not delete the authoritative configuration.

A Session token protects the viewer from writing into a different Session if GX-TXT changes Session while an external viewer process is still open.

14. Notes

The Notes tab stores UTF-8 configuration notes suitable for scientific context such as legend interpretation, scan distortion, or excluded dashed curves. Changes are retained in the Session Interactive Document.

When the user publishes curve Datasets or an enhanced image Dataset, nonempty configuration notes are also appended as a clearly titled section of each published Dataset's **Dataset Notes** Markdown document (notes/datasets/*.md). The existing Dataset Notes content is preserved, and publishing the same notes again does not duplicate the section. If the configuration text changes, a new note section preserves the earlier published record rather than silently deleting it. Empty configuration notes leave existing Dataset Notes unchanged.

These Markdown notes are distinct from structured Dataset metadata and from the short Dataset description shown in the Registry. Use the framework's Dataset Notes editor or Session Files viewer to inspect the resulting .md document. If a notes update fails after scientific publication, the viewer reports that the Dataset was published but its notes were not saved; the user may repeat the publication after resolving the issue.

The Notes tab is not intended as a general text-editor subsystem.

15. Publishing independent curves

With numerical-family mode disabled, Curve Digitalizer publishes `table.generic` data in long form. The **Publish** tab lets you select one or more accepted calibration measurements and choose **separate** or **combined** publication. See the publication layouts and naming section below.

Current stored variable IDs (not part of the consumer contract) are:

Variable ID	Meaning
x	calibrated X value
y	calibrated Y value
curve_id	stable internal curve identity
curve_label	user-visible curve label
point_index	zero-based order within the curve

X and Y carry the configured labels, units, semantic roles, value kind/dtype, and canonical unit when Unit Registry resolves one.

The module does not sort X globally. Non-monotonic curves remain non-monotonic.

15.1 Semantic column contract

`table.generic` alone does not explain the meaning of its columns. The output therefore includes the explicit Dataset Metadata fields:

```
curve_digitizer.curve_schema_version=1
curve_digitizer.curve_mode=independent
curve_digitizer.curve_grouping=curve_key
curve_digitizer.curve_row_dimension=row
curve_digitizer.curve_column.x=x
curve_digitizer.curve_column.y=y
curve_digitizer.curve_column.curve_key=curve_id
curve_digitizer.curve_column.curve_name=curve_label
curve_digitizer.curve_column.point_order=point_index
```

Each mapping's right-hand side is the current Variable ID, **not** its semantic meaning. Producers may change those technical IDs in future releases without breaking consumers that use `CURVE_DIGITIZER_DATASET_SCHEMA_API_v1`.

Stable semantic role	Meaning
x	Calibrated X coordinate, with unit from Variable Schema
y	Calibrated Y coordinate, with unit from Variable Schema
curve_key	Opaque, stable curve identity; group by this , not by curve label
curve_name	Editable human-readable title; never use as identity

Stable semantic role	Meaning
point_order	Zero-based point insertion order within curve_key
family_parameter	Repeated numeric family parameter, only in numerical-family mode

Rows for different curve_key values are independent curves even when stored in one Dataset. The curve_key is only unique within the published Dataset; it is not an externally global identifier. A change to a curve name does not change its curve ID. X need not be monotone and must not be used to reconstruct order.

15.2 Public consumer API

```
set manifest [::GX::CurveDigitizerDataset::read $datasetId]
set mode [dict get $manifest mode]
set members [dict get $manifest columns curve_key]
set points [dict get $manifest columns point_order]
set xy [list [dict get $manifest columns x] [dict get $manifest columns y]]
set display [dict get $manifest columns curve_name]
# For a true family only:
if {$mode eq "numerical_family"} {
    set parameter [dict get $manifest columns family_parameter]
}
```

The manifest returns the corresponding V3 variable descriptors and textual role meanings, verifies that every mapped variable exists, checks dimension and dtype, and rejects malformed or inconsistent schema/contract pairs. It does not read private Interactive Documents and does not require a running Curve Digitalizer GUI. A convenience column \$datasetId \$semanticRole returns a single actual Variable ID; hasSchema and schema expose availability and the public contract. See CURVE_DATASET_SCHEMA_API_v1.md for the complete normative spec.

For historical published Datasets without the explicit semantic schema, the API provides a marked **legacy** mapping only when the record is genuinely Curve-Digitizer-owned and all original technical Variable IDs can be verified in Schema v3. It never infers scientific meaning from a human-readable column label.

16. Publishing a numerical curve family

When numerical family mode is enabled, **Publish Dataset** writes the existing GX-TXT curve_family.table contract.

Current stored variable IDs (not part of the consumer contract) are:

Variable ID	Meaning
x	independent coordinate
family_value	numerical family parameter
y	measured/dependent value
curve_id	stable internal curve identity
curve_label	user-visible curve label

Variable ID	Meaning
point_index	zero-based order within the curve

The default semantic roles are independent, family, and measured, but the user-visible configuration remains explicit.

Numerical-family metadata sets `curve_digitizer.curve_mode=numerical_family` and adds `curve_digitizer.curve_column.family_parameter=family_value`. All five mandatory roles are still present and points are **still grouped by** `curve_key`. A family value does not replace the grouping key.

17. CSV artifact and Canonical Dataset

Each output Dataset is committed through GX-TXT Dataset Publication API v2:

```
Publish selected calibrations
-> verify source image SHA-256 and Session identity
-> validate selected measurement(s), schema and Dataset names
-> publish an individual scientific Dataset transaction
-> CSV scientific artifact, Variable Schema and Dataset Metadata
```

When separate publication selects multiple calibrations, each Dataset is committed separately. If a later publication fails, earlier completed outputs are preserved and the viewer explicitly reports which Dataset failed. This is not an all-or-nothing multi-Dataset transaction.

The user does not export a CSV and then import it again through Acquisition.

The CSV remains a portable scientific artifact, while the Canonical Dataset is immediately available to compatible GX-TXT consumers.

18. Calibration selection, publication layout and Dataset names

The **Publish** tab shows a scrollable list of **accepted** measurements. Select one or several measurements using standard multi-selection keys. The selection is independent of the measurement currently **Active** in Calibration: activating a measurement changes the on-screen coordinate mapping, not which measurements are selected for publication. Draft and rejected measurements are preserved but are not offered for publication until explicitly accepted.

Two layout policies are available:

Layout	What is published	Compatibility
separate	One independent Dataset per selected calibration, each with X/Y calculated from its own transform; all curve points keep original raster X/Y	Recommended for ordinary Plot/Fit consumers; no calibration-key column required
combined	One Dataset with one realization of each raster point for every selected calibration; each row carries <code>calibration_key</code> and <code>calibration_name</code> roles	Only for consumers that understand calibration-key grouping; generic X/Y readers may accidentally mix realizations

The default is **separate**. Choose the default under **GX-TXT → Configuration → Modules → Curve Digitalizer → Calibration publication layout**. The same choice is accessible from **Configuration → Viewer settings → Axes & output**, where Apply creates a Session override using CORE Configuration. The Publish tab may temporarily select another layout for its current configuration. Both `table.generic` and `curve_family.table` preserve their curve semantics, but combined output adds another mandatory grouping dimension for calibration-aware consumers.

18.1 Template-based Dataset IDs

The Publish tab's **Publication base Dataset ID** is the user-defined `[name]` token. A format string named **Dataset naming template**, edited via GX-TXT module configuration or **Viewer settings → Axes & output**, constructs the actual Dataset IDs. The default is:

```
[name].[calibration].curves
```

Tokens are case-sensitive and enclosed in square brackets:

Token	Substitution
[name]	Exact validated user-provided base Dataset ID
[calibration]	Measurement label converted to a safe ASCII Dataset-ID component (or combined for a combined publication)
[calibration_id]	The stable measurement_id, sanitized for Dataset IDs (or combined)
[configuration]	Human configuration label, sanitized
[index]	Position of the Dataset in the selected publication plan (starting at 1)
[mode]	separate or combined

A naming template **must** include `[name]`; only these tokens and ASCII letters, digits, dots, underscores and hyphens are accepted. The resulting ID must begin with a letter and be at most 128 characters. Invalid tokens, missing required values, and duplicate output names are rejected **before** publication. The Publish tab shows a preview of the actual generated Dataset IDs.

For example, a base ID `sem.grooves` with two accepted measurements named `Trial A` and `Trial B` produces:

```
sem.grooves.trial_a.curves
sem.grooves.trial_b.curves
```

The combined layout instead produces `sem.grooves.combined.curves`. If two measurements have the same label and the default template would collide, rename one of them or include `[calibration_id]` or `[index]` in the template.

18.2 Source lineage, identities and republishing

Each output Dataset contains the complete group of calibration measurements in its structured Dataset Metadata. A separate result identifies the calibration that generated its X/Y columns; a combined result identifies **all** selected calibrations and maps rows through the `calibration_key` semantic role. The source image identity, transform fingerprints and original raster points are preserved. This allows downstream programs to recover other realizations without depending on the live Interactive Document.

Publishing the same naming plan again may replace the existing Dataset only under GX-TXT's own-producer replacement rules and only when its recorded configuration identity matches the current digitizer configuration. Publishing never silently overwrites another configuration's scientific output. A user may inspect or change the names in the Publish tab before executing.

Combining calibrations requires consistent units for all X variables and all Y variables. This is verified before writing rows. Every selected calibration must validly map **every** point; the publisher rejects a missing/out-of-domain point rather than silently omitting it. No statistical average, uncertainty or interpolation is manufactured by either mode.

19. Provenance and metadata

Published results record lineage through the normal GX-TXT mechanisms, including information equivalent to:

- `producer viewer.curve_digitizer;`
- plugin version;
- configuration Interactive Document ID;
- configuration name;
- source Image Dataset ID;
- source SHA-256;
- source Dataset relationship;
- calibration type;
- X/Y scale modes;
- X/Y labels and units;
- family label/unit when applicable;
- publication contract;
- publication timestamp.

Source image metadata remain attached to the source Dataset. Curve Digitalizer does not blindly duplicate every source metadata key into the output Dataset.

20. Using the result with `plot.dataset`

The published Dataset is independent of the viewer. A compatible `plot.dataset` PlotSpec can bind directly to the stable x and y variables of a generic result.

Curve Digitalizer private state is not required for plotting.

21. Using the result with fitting modules

For a genuine curve family, the published `curve_family.table` Dataset exposes the current GX-TXT contract plus the expected semantic roles for independent coordinate, family parameter, and measured value.

Curve-family/nonlinear fitting consumers can therefore select the published Dataset through their normal Dataset contract/binding logic. They must not read Curve Digitalizer private configuration documents.

22. Errors and recovery

No open Session

Open a GX-TXT Session before opening Curve Digitalizer.

No compatible Image Dataset

Import an image through the normal GX-TXT acquisition/image importer first.

Desktop dependency error

Install/enable Python 3, Tkinter, NumPy, and Pillow/ImageTk. Plugin discovery remains usable without them.

Calibration invalid

Check point order, non-degenerate plot geometry, axis limits, and positivity of logarithmic-axis limits.

Stale source

The source image bytes changed. Do not reuse the old calibration blindly. Restore the exact source or create/reset a configuration explicitly for the new image.

Missing source

The saved Dataset ID no longer resolves. Restore/re-import the intended source as appropriate. The configuration itself remains stored.

Family publication rejected

Ensure all curves have numerical family values and consistent family units.

Publication ownership error

Choose another output Dataset ID if the requested ID belongs to another producer.

Session changed while viewer was open

Close the stale Curve Digitalizer window and reopen it from the intended current Session.

23. Known limitations

- The viewer is manual/semi-assisted; it does not implement automatic curve tracing or OCR.
- A source must already exist as a compatible Image Dataset; direct filesystem image opening is intentionally absent.
- The interactive adapter requires a working Tk display (local desktop, X11 forwarding, or equivalent).
- Optional darkest-pixel snapping is local and image-intensity-based; it does not understand curve semantics.
- Scientific downstream execution depends on the downstream module/runtime. For example, actual `plot.dataset` rendering or numerical fitting requires the runtime used by those modules (normally GNU Octave on the target GX-TXT installation).
- The plugin does not modify CORE and does not provide its own private image importer, unit registry, worker pool, or publication store.

24. Keyboard/mouse interaction

Core image interaction inherited from the original canvas includes:

F1	Open integrated detailed user guide
Mouse wheel	Zoom
Middle drag	Pan
Right drag	Pan
F	Fit image to window
Ctrl+Z	Undo
Ctrl+Y	Redo
Delete	Delete selected point where applicable

Standalone shortcuts for file/project/export operations are not part of the GX-TXT-native viewer.

25. Technical state format

The Session document format is:

```
gtxt.curve_digitizer.configuration
```

It is an implementation state contract for this plugin, not a replacement for published scientific Datasets. Downstream modules must consume the published Dataset rather than this document.

Companion-module integration

Curve Digitalizer provides `CURVE_DIGITIZER_COMPANION_API_v1` for tightly coupled tools such as Curve Trace Assistant. The authoritative state remains the existing `gtxt.curve_digitizer.configuration` Interactive Document. Companion modules do not create their own project file or duplicate calibration.

The API exposes stable configuration and curve identities, resolves the associated Image Dataset through public Dataset APIs, verifies the stored source SHA-256, and can append accepted pixel points to a selected

curve. Pixel-to-scientific conversion is performed by the same Curve Digitalizer calibration model used by manual points.

Companion writes are guarded by both the live Session token and the expected Interactive Document SHA-256. If another window or module changed the configuration, the write is rejected and the caller must reload.

26. Published calibration provenance

On publishing an output Dataset, Curve Digitalizer includes the calibration manifest in the **same atomic Dataset Publication API v2 transaction** as the scientific data. Output metadata is not written after finalization. The metadata travels with the published Dataset, survives Session reopen and remains available to independent viewers.

Fields under `curve_digitizer.*` include `source_dataset_id`, `source_sha256`, `configuration_id`, `calibration_type`, `image_width`, `image_height`, `coordinate_origin`, `y_direction`, `physical_unit`, and `transform`; optional `scale_bar` and `uncertainty` are present when applicable. The transform includes an exact raster-to-normalized 3×3 homography and the linear or logarithmic scientific axis maps. For two linear axes, an additional raster-to-scientific homography is published. The physical unit is normalized to a canonical code (μm in the GUI becomes `um` for the manifest).

Calibration types: `pixel` for unmodified full-image pixel axes, `scale_bar` for uniform full-image physical scaling, `affine` for a linear affine four-corner map, and `projective` for genuinely perspective four-corner transformations. A log axis is represented by a normalized homography plus explicit non-linear axis mapping, not by a false linear scientific matrix. No uncertainty is invented if none has been supplied. The scale-bar reference records measured endpoints in original image pixels, declared physical length, canonical unit, original UI unit, and selection method.

After same-owner publication to an existing output Dataset, unrelated metadata fields are retained. Obsolete `curve_digitizer.*` fields from a previous calibration are replaced, not silently retained.

27. Public API for other viewers

The registered API ID `CURVE_DIGITIZER_CALIBRATION_API_v1` has its own module entrypoint and can resolve calibration **using the published output Dataset ID alone**. Its read function uses the CORE public Dataset Metadata API and Session Inspection API and does not read the viewer's private Session document.

```
set manifest [::GX::CurveDigitizerCalibration::read digitized.sem_contour]
set transform [dict get $manifest transform]
set H [dict get $transform normalized_homography]
set source [dict get $manifest source_dataset_id]
set unit [dict get $manifest physical_unit]
if {[dict exists $manifest scale_bar]} {
    set measured [dict get [dict get $manifest scale_bar] pixel_length]
}
```

See `CALIBRATION_API_v1.md` for the exact contract, optional Session override, schema discovery, unit conventions, error behavior, and formulae.

28. Status bar visibility

The main status bar is anchored/reserved at the bottom of the viewer ahead of the resizeable central workspace. The initial window height also adapts to the screen instead of assuming that an 880-pixel client area fits on every desktop. Long Calibration panels may need available vertical space, but must not obscure status information.

29. Menu bar and viewer settings

The narrow top row now contains only the active **Configuration** and **Source Image Dataset** selectors, including the source integrity indicator. Viewer commands are grouped into native pull-down menus:

Menu	Commands
Edit	Undo (Ctrl+Z), Redo (Ctrl+Y), New/Edit/Delete curve, Delete selected point
View	Fit image, jump to Source/Calibration/Curves/Publish tabs; show point numbers, scale overlay and calibration corners
Calibration	Auto-calibrate full image, detect bar, manually select endpoints, apply physical scale, verify and clear calibration
Dataset	Open the Publish tab or publish directly (subject to existing safety checks)
Configuration	New/Rename/Duplicate/Delete a Session configuration; Viewer settings...
Help	Curve Digitalizer user guide... (F1), About

29.1 Viewer settings window

Open **Configuration** → **Viewer settings....** Settings are grouped in five pages:

1. **Display** — Show point numbers; show or hide the saved scale-reference overlay and calibration-corner markings. Candidate scale-bar previews stay visible even if saved overlay display is disabled.
2. **Digitizing** — Optional snap to the darkest pixel and snap radius, from 0 to 25 source pixels.
3. **Calibration** — Default scale-bar tick reference (center, inner, outer), full-image Y direction and new scale-bar measurement defaults (positive length and valid physical unit). These are preferences for subsequent calibration actions; changing them does not rescale a previously accepted measurement.
4. **Axes & output** — Displays the currently configured axis labels/units, provides a jump to the scientific axis editor, and allows changing the output Dataset ID and display name without publishing anything.
5. **Configurations** — Lists all saved configurations in the current Session and highlights the active one. Switch or manage them using the main Configuration selector/menu. Deleting a configuration retains its existing confirmation workflow.

Apply settings validates the entries and commits changed viewer preferences through GX-TXT Configuration API. User defaults and Session overrides follow the framework's normal precedence. Scientific axes and output

identity remain part of the Interactive Document. **Cancel** discards unsaved drafts. Applying settings never publishes a Dataset or modifies the source image.

29.2 Integrated searchable help

Open **Help → Curve Digitalizer user guide...** or press **F1**. The Help command asks the GX-TXT Tcl host to open a dedicated **rendered** manual window. The host retrieves the installed module's `MANUAL.md` through the framework Documentation Browser registry (`documentationBrowserRefresh / documentationBrowserRead`) and displays its Markdown/GX-Markup via the framework `markupRenderStringToWidget` renderer, not a Python source-text widget. Headings, emphasis, code blocks, lists and supported links follow GX-TXT's markup conventions. The manual remains read-only; the left contents panel jumps to chapters and **Find next** searches text without case sensitivity. The viewer-to-host request is asynchronous, so Help does not block the viewer's Tk event loop. No internet connection, external browser or CORE changes are required. If the registered manual or GX-Markup services are missing, the viewer reports the host's error instead of silently showing unformatted text.

30. Choosing SEM scale-bar tick geometry

A typical SEM scale bar is a horizontal shaft bounded by short vertical ticks. **The default is the distance between the centers of those two terminal ticks.** This is a convenient geometric assumption, **not a guarantee about the microscope manufacturer's convention.** A specific SEM annotation may use inner-to-inner or outer-to-outer tick edges. Consult the instrument exporter or its original metadata if the distinction materially affects precision.

Choose the tick reference in **Calibration → B · Scale bar**, or set its default in **Configuration → Viewer settings... → Calibration**:

- **Center-to-center (tick centers):** measure the distance between the centers of the terminal marks (default for new or migrated configurations).
- **Inner edges of ticks:** measure the gap between the inner-facing edges.
- **Outer edges of ticks:** measure the full extent between outer-facing edges.

Detect first finds the horizontal shaft, then searches near its ends for distinct upright tick segments. The viewer refuses ambiguous or low-contrast tick geometry rather than claim a center that was not detected. For photographs with indistinct ticks, use **Endpoints** and click the desired reference points directly. The preview line must be visually inspected before **Apply scale**. Automatic measurement is a convenience; its success alone does not certify metrological accuracy.

The chosen applied convention is recorded in `scale_reference.edge_convention` within the Session configuration and in the published Dataset Metadata key `curve_digitizer.scale_bar.edge_convention`. The stable `CURVE_DIGITIZER_CALIBRATION_API_v1` exposes it as `dict get [dict get $manifest scale_bar] edge_convention`. Historical Dataset manifests lacking the field are reported as `unspecified`; never silently reinterpret historical measurements as center-to-center. The existing uncertainty metadata remains optional: choosing a convention does **not** generate a numerical uncertainty estimate.

Compact header, namespace choice and scientific raster points

The header shows only **Configuration** and **Image Dataset**. Configuration names and Image Dataset IDs remain visible and editable through their existing combobox selectors. Select **Namespace → Select Dataset namespace...** for a scrollable, canvas-based list of GX-TXT Dataset namespaces. Selecting one filters the available image choices; it does not move, replace or relabel a Dataset. If the current configuration references an image outside the chosen filter, the original source reference is preserved. Use **Dataset → Refresh Dataset catalog** when necessary; automatic checks remain enabled by default and configurable in **Configuration → Viewer settings → Dataset browser**. Catalog revisions and check failures are displayed there. Source changes, missing sources and Session changes remain signaled in the bottom status bar and continue to block unsafe publication without losing digitized points.

All newly published digitized curves contain scientific X/Y and independent **original raster pixel** X/Y coordinates per point. Both the scientific calibration and original pixel positions remain available after deleting the Digitizer configuration document. The output Dataset is standalone; consumers should query the published calibration/schema contracts instead of assuming hard-coded Dataset variable labels.

Other viewers may use the following stable public calls:

```
set cal [::GX::CurveDigitizerCalibration::read $datasetId]
set identity [::GX::CurveDigitizerCalibration::identity $datasetId]
set coords [::GX::CurveDigitizerCalibration::transformPoints \
    $datasetId {{150.5 90}} {200 140}} pixel_to_scientific]
set jac [::GX::CurveDigitizerCalibration::jacobian $datasetId {{150.5 90}}]
set uncertainty [::GX::CurveDigitizerCalibration::pointUncertainty \
    $datasetId {{150.5 90}}]
set curveSchema [::GX::CurveDigitizerDataset::read $datasetId]
set points [::GX::CurveDigitizerDataset::rasterPoints $datasetId "" 0 10000]
```

The producer uses the *pixel center* convention. Unknown upstream image preprocessing or unknown measurement uncertainty is recorded as unknown, not as an exact scientific measurement. Further details, supported mappings, noninvertible transformation errors and optional uncertainty propagation are specified in CALIBRATION_API_v1.md and CURVE_DATASET_SCHEMA_API_v1.md.

GX-TXT module configuration

The Curve Digitalizer registers its presentation, digitizing, calibration **defaults** and Dataset catalog preferences through CONFIGURATION_API_v1. In GX-TXT open **Configuration → Modules → Curve Digitalizer** to edit global **User defaults** or **Session overrides**; inherited settings use the standard CORE precedence. Alternatively use **Configuration → GX-TXT module configuration...** in the viewer, which opens the standard CORE module editor. **Configuration → Viewer settings...** writes the shared fields as Session overrides when Apply is pressed. Invalid values and stale drafts are rejected by CORE as a whole transaction. An open viewer checks for central configuration changes every five seconds.

Shared preferences include point numbers, scale/corner overlays, snapping, scale-bar conventions, Y direction, default physical scale, automatic catalog checks, and verification marker shape/color/size/line width. The verification marker is an X; its configurable properties are verification_marker_size, verification_marker_width and verification_marker_color. Scale-bar settings are defaults for new detections, **not** retroactive changes to an existing scientific calibration. Scientific axes, measured scale references, digitized points, names, publication targets and source bindings remain in CORE Interactive Documents; they do not belong in global module preferences. Legacy documents continue to load, but their previous UI preference snapshots do not override the effective GX-TXT Configuration registry.

Calibration tab: layout and scientific scope

The Calibration tab places **A • Full image** beside **B • Scale bar**. These are two related operations: full-image calibration uses all source pixel centers, while a scale-bar measurement converts distances to real physical units. They are distinct from manual **four-point projective calibration**, intended for plots with axes and possibly perspective distortion.

B • Scale bar → Clear selection overlay hides the saved scale-bar line, endpoints and label from the canvas, without changing the physical calibration, the raster points or published coordinates. Starting a new measurement replaces an earlier candidate. The saved reference can be displayed again by reopening the configuration; **Calibration → Reset physical scale...** explicitly resets physical scale back to full-image pixel coordinates after confirmation.

Repeated calibration measurements

Calibration measurements is a full-width, scrollable table below the Projective calibration and verification controls. It provides independent calibration realizations for the same source raster. The **New** control saves the current realization, creates a new one with unchanged axes but no selected corners, and leaves the digitized curve points untouched. Select each of the four corners independently. You can also auto-calibrate a full image or apply a scale bar when scientifically appropriate.

- **Activate** switches the current mathematical mapping and recomputes the scientific coordinates of existing source-raster curve points. Pixel positions and curve identities are never edited by switching calibrations.
- **Accept** checks that the selected calibration has valid corners and axes and marks it suitable for downstream study. Any accepted measurement may be selected in the Publish tab, even if it is not active.
- **Reject** retains the realization and its data while marking it unsuitable for the default accepted set. Rejection does not erase it.
- **Cancel draft** discards only an unfinished draft with confirmation, leaving accepted measurements untouched. At least one measurement must remain.
- **Rename** changes the measurement's display label without changing its stable measurement identity or transform fingerprint.
- **Delete** removes the selected measurement only after confirmation; at least one realization must remain. Published Datasets are not silently changed.

Each measurement has a stable `measurement_id` that identifies the operator's particular realization, even if its numeric transformation is identical to another. The group has a persistent `calibration_group_id`. A separate `calibration_sha256` fingerprints the *mapping*, so equality of the hashes means identical raster-to-scientific mapping, not identical measurement IDs.

The Interactive Document stores all measurements and the currently selected one. On Dataset publication, their statuses, original corner pixels, axis ranges and, when valid, individual transformations and fingerprints are copied into the published Dataset metadata. Scientific X/Y columns are calculated from the **selected** calibration, rather than necessarily from the active one. For combined publication, each calibration generates its own numeric rows and `calibration_key` distinguishes them. All source-raster points remain available. No automatic variance, standard deviation, or uncertainty is fabricated from repeated clicks.

Other modules can inspect the published measurements through:

```

set measurements [::GX::CurveDigitizerCalibration::replicates $datasetId]
set firstId [dict get [lindex $measurements 0] measurement_id]
set points [::GX::CurveDigitizerDataset::rasterPoints $datasetId]
set output [::GX::CurveDigitizerCalibration::transformPoints \
    $datasetId {{120 150} {210 180}} pixel_to_scientific $firstId]

```

transformPoints accepts either an explicit measurement ID or no ID to use the active published mapping. For incomplete measurements it reports an error; unknown IDs are rejected. Calibrations must refer to the same source raster and valid scientific domain. To estimate manual calibration repeatability, external analysis software can apply the accepted transformations to the *same* pixel coordinates and then calculate variability. Repeated calibration cannot by itself establish absolute instrument accuracy or systematic error.

Calibration verification

The verification panel is beside the four projective-calibration corner buttons; calibration measurements occupy the scrollable section underneath.

Select **Verify calibration...** to reveal the Expected X / Expected Y fields. While in verification mode, clicking the original image places a bright red **X** exactly at the selected pixel-center location. The marker is temporary, with a configurable size, color and line width in *screen pixels*, independent of zoom. The program displays the calculated scientific coordinates and, when provided, the differences from the expected values. You may enter or change expected values *after* selecting the image point; the residuals update immediately and the marker remains visible.

A subsequent click moves the X instead of accumulating markers. Press **Esc**, choose **Clear verification**, or switch to another image-editing mode to remove it. No temporary verification marker is saved in a Dataset or Interactive Document. A verification residual is not automatically a measurement uncertainty estimate.

Image Processing: non-destructive enhancement

Open **Image Processing** in the viewer's left tab bar (or **View → Image Processing...**). The original canonical image is always preserved. ImageMagick is resolved by GX-TXT **Configuration → Runtime → ImageMagick**, not from a hard-coded path. The viewer remains fully usable when ImageMagick is absent; only image processing and publication of the processed image are unavailable.

Filter controls, processing order and preview

Image Processing presents one row for each supported filter. Enable the desired operation with its checkbox, then adjust the numeric sliders, spinboxes or choice selectors. A disabled filter does not participate in processing. A numeric filter retains its value when disabled and can be enabled again.

Filter	Controls	Intended use
Color channel	Red / Green / Blue / Luminance	Choose an informative component
Grayscale	Checkbox	Remove chromatic differences
Median denoise	Radius slider, 1..10	Suppress small-scale noise
Auto levels	Checkbox	Expand intensity distribution

Filter	Controls	Intended use
Contrast stretch	Tail percentage slider, 0..20	Suppress near-extreme tails
Brightness / contrast	Separate sliders, -100..100	Adjust intensity and contrast
Gamma	Slider or numeric spinbox	Nonlinear tonal adjustment
CLAHE	Tile grid, histogram bins and clip value	Local contrast
Sharpen	Sigma slider, 0.1..4.9	Improve perceived boundaries
Invert colors	Checkbox	Reverse light/dark appearance
Threshold	Percentage slider, 0..100	Segment foreground/background

The **Standard** filter-order mode applies enabled operations in the documented, fixed order. This is a practical starting point, not a universal optimum. Channel selection and Grayscale are mutually exclusive in Standard mode.

Advanced exposes the exact ordered pipeline; choose a step and use **Move up**, **Move down**, or **Remove**. Existing repeated steps are preserved until explicitly modified. Changes in operation order change the output and are stored in the original order for reproducibility. Switching explicitly from Advanced to Standard sorts operations by the standard order.

Live preview is available when ImageMagick is configured. Slider updates are debounced: an obsolete processing result is discarded rather than applied over newer settings. **Apply processing / Preview** remains available for manual refresh. **Reset filters** clears the current processing pipeline.

Original / processed split comparison

Compare original and processed images enables a two-viewport view only while the Image Processing tab is selected. The original is on the left; the filtered image is on the right. Drag the vertical divider to assign more space to one viewport. Each viewport independently supports mouse-wheel zoom, right/middle-button drag pan and its own raster position and scale.

Synchronize pan and zoom optionally keeps both viewports at equal zoom, centered on the same original-raster coordinate even if the divider makes their displayed sizes different. Disabling synchronization retains independent navigation. On first activation, or after loading a new source image, each panel automatically fits its complete image to its actual available viewport area. Once initialized, switching tabs, adjusting the divider or changing filters does not reset manually chosen pan and zoom. A saved single viewer position is also preserved when restoring the Session. **Show digitized curve overlays** draws point markers in both viewports as read-only indicators. The image comparison NEVER adds, moves, or deletes curve points.

Switching from Image Processing to another tab restores the usual single image canvas. The processed image may still be displayed for easier tracing, but a point can be inserted, dragged or deleted **only while Curves is the active tab**. Image Processing, Source, Calibration, Publish and Notes never alter curve points through canvas interaction or the Delete shortcut. The Calibration tab retains only its explicit corner/scale/verification actions.

The original Image Dataset and raster coordinate mapping remain authoritative. The processing pipeline never changes the calibration matrix, raster dimensions or original-image SHA-256. Switching the preview does not rewrite points.

The derived processing stream is based on the viewer's decoded RGB image; the output is RGB PNG rather than a bit-exact copy of higher-bit-depth acquisition channels. Keep the original canonical Dataset for quantitative image metrology.

ImageMagick may emphasize or suppress features and can alter the *apparent* location of an edge. Inspect the original before making a final measurement. There is no crop, resize, rotation, flip or perspective correction in this viewer. Use a dedicated image-transformation viewer for geometry-changing operations.

Publish processed Image Dataset

1. Select and preview at least one filter.
2. Enter the new **Image Dataset ID** and optional **Image description**.
3. Click **Publish image.raster Dataset...** and review the source and output.
4. The viewer publishes a standalone canonical image.raster PNG via CORE's DATASET_FILE_PUBLICATION_API_v1 and refreshes the Dataset catalog.

The description is stored in the Dataset notes; the canonical Dataset ID is the stable output identity. The original Dataset is never overwritten. An image published here can be opened by other image-compatible viewers, including Curve Digitalizer in a **new digitizer configuration**. No curve Dataset is created by this action; **Publish selected calibrations** is a separate, independent operation. Processing state is saved with the digitizer's Session Interactive Document. The newly published Image Dataset holds its own lineage and processing manifest, so it remains interpretable when that document is unavailable.

Processed-image metadata and scientific traceability

The output contains the canonical parent Dataset ID, human-readable source name, parent SHA-256, ordered processing steps with exact validated parameters, ImageMagick version, full raster width/height, geometry-preserved assertion, identity pixel transform and a deterministic processing fingerprint. CORE supplies the authoritative SHA-256 for the published output bytes. Data registration and structured metadata commit occur in the same transaction.

These structured fields are **not** intended for manual parsing by downstream viewers or scientific scripts. Use the public API described in the Developer Reference below.

Developer Reference — Public APIs

These APIs are stable module-owned interfaces of the GX-TXT CORE Curve Digitalizer plugin. Tcl viewers call registered `::GX::*` commands; user scripts use the *same public scientific contracts* through the registered Python/Octave Script Bridge provider operations. API ID suffixes are formal contract identifiers, not viewer-release numbers. All methods operate on canonical Dataset IDs.

CURVE_DIGITIZER_CALIBRATION_API_v1

Read physical calibration, original-image identity, scale-bar geometry and calibration realizations without knowing any metadata storage key.

```
set c [::GX::CurveDigitizerCalibration::read $datasetId]
set trials [::GX::CurveDigitizerCalibration::replicates $datasetId]
set xy [::GX::CurveDigitizerCalibration::transformPoints \
  $datasetId $rasterPoints pixel_to_scientific]
```

The transform also accepts `scientific_to_pixel` and a selected calibration realization where available; invalid/invertible-domain violations produce explicit errors. Uncertainties are returned only when measured, not invented. See `CALIBRATION_API_v1.md` for precise argument forms and values.

CURVE_DIGITIZER_DATASET_SCHEMA_API_v1

Resolve scientific columns using semantic roles rather than names or positions:

```
set schema [::GX::CurveDigitizerDataset::read $datasetId]
set curveKey [dict get $schema columns curve_key]
set xId [dict get $schema columns x]
set yId [dict get $schema columns y]
```

Additional roles include curve label, point order, original raster X/Y and, in combined calibration publications, calibration key/name. Independent curves and numerical families have distinct schemas and grouping rules. Use `CURVE_DATASET_SCHEMA_API_v1.md` for the full contract.

CURVE_DIGITIZER_IMAGE_PROCESSING_API_v1

The processed-image facade presents the original image, ordered filters, processing engine, raster identity and entire verified image lineage:

```
set p [::GX::CurveDigitizerImage::read $datasetId]
set steps [::GX::CurveDigitizerImage::operations $datasetId]
set parents [::GX::CurveDigitizerImage::lineage $datasetId]
set ok [::GX::CurveDigitizerImage::hasManifest $datasetId]
```

The returned `p` includes `source_dataset_id`, `source_display_name`, `source_sha256`, `processing_operations`, `processing_engine`, `processing_engine_version`, `geometry_preserved`, `pixel_transform`, `image_width`, `image_height`, `processing_fingerprint` and `output_sha256`.

Python Script Bridge example:

```
from gtxt import gtxt_api_call, gtxt_api_require

gtxt_api_require('CURVE_DIGITIZER_IMAGE_PROCESSING_API_v1@1:viewer.curve_digitizer')
info = gtxt_api_call('CURVE_DIGITIZER_IMAGE_PROCESSING_API_v1', 'read',
  ['sem.enhanced'])
print(info['source_dataset_id'], info['processing_operations'])
```

GNU Octave Script Bridge example:

```

if ~gtxtxt_api_has('CURVE_DIGITIZER_IMAGE_PROCESSING_API_v1')
    error('Curve Digitalizer image provenance API unavailable');
endif
info = gtxtxt_api_call('CURVE_DIGITIZER_IMAGE_PROCESSING_API_v1', 'read',
{'sem.enhanced'});
disp(info.source_dataset_id);

```

`read(dataset_id)` returns a structured manifest. `operations(dataset_id)` returns ordered {kind,value} records. `lineage(dataset_id)` returns the SHA-verified parent chain up to the untouched original. `has_manifest(dataset_id)` is exposed by Script Bridge; its Tcl equivalent is `hasManifest`. Missing, corrupt and incompatible metadata generate explicit errors. Consumer code must not parse `.ini` files or rely on the names of internal metadata fields.

A full description of each operation, result, error and bridge behavior is in `IMAGE_PROCESSING_API_v1.md`. Other viewer APIs are documented in their respective public API references. The integrated Help window renders this Markdown with GX-TXT's standard GX-Markup renderer.

APPENDIX B — Complete public API specifications

The following three specifications are rendered from the exact documentation files distributed with the Curve Digitalizer plugin. They complement the user-facing `MANUAL.md` by describing machine-readable contracts, argument conventions, metadata validation and Script Bridge consumers. They are included here to make this single PDF usable as a complete technical reference.

CALIBRATION API v1.md

CURVE_DIGITIZER_CALIBRATION_API_v1

SPDX-License-Identifier: GPL-3.0-or-later

Public provider

- API ID: `CURVE_DIGITIZER_CALIBRATION_API_v1`
- Module: `viewer.curve_digitizer` (core-distributed module, not a CORE extension)
- Tcl entrypoint: `api/curve_digitizer_calibration_api_v1.tcl`
- Namespace: `::GX::CurveDigitizerCalibration`
- Minimum GX-TXT baseline: existing public Dataset Metadata v1, Session Inspection, Dataset Publication v2

This API resolves calibration by the **published curve Dataset ID**. It does not inspect image files, read private digitizer configuration documents, depend on a currently open viewer window, or parse plugin-specific metadata in consumers.

```

set info [::GX::CurveDigitizerCalibration::read digitized.sem_contour]
set unit [dict get $info physical_unit]
set transform [dict get $info transform]
set H [dict get $transform normalized_homography]
set dimensions [list [dict get $info image_width] [dict get $info image_height]]
set hasBar [dict exists $info scale_bar]

# Optional Session override (defaults to current Session):
set info [::GX::CurveDigitizerCalibration::read digitized.sem_contour $session]

# Non-throwing predicate for a Dataset with a v1 calibration manifest:
set available [::GX::CurveDigitizerCalibration::hasCalibration digitized.sem_contour]

# API declaration/discovery:
set contract [::GX::CurveDigitizerCalibration::schema]

```

A Dataset absent from the Session, not produced by Curve Digitalizer, missing required calibration metadata, or declaring an unsupported manifest version raises an explicit error in `read`. The predicate returns 0 for unavailable or unsupported Dataset metadata; `read` remains authoritative for validation.

Stable return fields (v1)

Tcl dict field	Meaning
api_version	Always integer 1
manifest_version	Always integer 1
dataset_id	The published scientific curves Dataset ID
source_dataset_id	Exact original image Dataset ID
source_sha256	SHA-256 digest of the linked source image bytes
configuration_id	Stable digitizer configuration identifier, independent of display name
calibration_type	pixel, scale_bar, affine, projective
image_width, image_height	Original raster dimensions in pixel centers, integer counts
coordinate_origin	bottom_left or top_left for scientific coordinates
y_direction	up or down in raster coordinates
physical_unit	Canonical physical length code (um, nm, mm, ...), or empty when none / mixed axes
transform	Structured transform dictionary, below
scale_bar	Optional structured bar reference when the calibration type is scale_bar
uncertainty	Optional flat structured uncertainty dictionary when present in source configuration
uncertainty_json	Optional canonical JSON uncertainty if supplied, including nested payloads

Missing uncertainty means **unknown**, not zero. Do not use homography corner-fit RMS (zero for four exact control points) as a measurement-uncertainty estimate.

transform structure

Field	Meaning
convention	column_vector_homogeneous_xy1
raster_origin	top_left always
pixel_reference	pixel_center always
normalized_coordinates	x_left_to_right_y_bottom_to_top
normalized_homography	A 3×3 Tcl list of lists, raster pixel centers → normalized coordinates (u,v)
x_axis, y_axis	Tcl dictionaries with minimum, maximum, scale (linear / log) and original axis unit
scientific_homography	Optional 3×3 raster → scientific coordinate matrix, present only when both axes are linear

Use homogeneous vectors: $q = H * (\text{pixel_x}, \text{pixel_y}, 1)$, then $(u, v) = (q_0/q_2, q_1/q_2)$. For linear axes, $x = x_{\min} + u*(x_{\max}-x_{\min})$ and $y = y_{\min} + v*(y_{\max}-y_{\min})$. For a logarithmic axis, use $x = 10 ** (\log_{10}(x_{\min}) + u*(\log_{10}(x_{\max})-\log_{10}(x_{\min})))$ (analogously for Y); its scientific_homography is intentionally absent.

For full-image autocalibration, raster coordinate (0,0) is the top-left pixel center, (W-1,H-1) the bottom-right pixel center. Physical dimensions use $(W-1)*\text{physical_length}/\text{pixel_length}$ and $(H-1)*\text{physical_length}/\text{pixel_length}$, not W or H.

scale_bar structure

When the type is scale_bar, the result provides start_px, end_px (2-element coordinate lists), pixel_length, physical_length, unit (canonical), display_unit (original UI spelling) selection_method (e.g. detected / manual) and edge_convention (center, inner, outer or unspecified for legacy publications). The endpoints are measured on the **original source raster**, not transformed display coordinates. The UI's persistent green overlay is a visualization, not the metadata authority.

Dataset Metadata storage

The authoritative public Dataset Metadata resource is managed by CORE DATASET_METADATA_API_v1. It contains the corresponding domain keys under curve_digitizer.* (e.g. curve_digitizer.source_sha256, curve_digitizer.calibration_type, curve_digitizer.transform). curve_digitizer.calibration_manifest and nested curve_digitizer.transform are canonical UTF-8 JSON strings for portable consumers. Scalar/matrix subkeys such as curve_digitizer.transform.h00 exist for a Tcl-only contract implementation with no JSON dependency. A caller should use the public Tcl API rather than depend on these internal representation details.

Metadata is included through the structured_metadata definition of **Dataset Publication v2** and therefore committed with the output Dataset; no post-finalization overwrite is needed. Unrelated metadata is kept

during same-owner republishes. Old Curve Digitalizer namespace values are removed before merging the new manifest, preventing stale scale-bar or uncertainty fields.

The API is part of the packaged Curve Digitalizer module. Consumers must declare its API dependency and handle a disabled/missing provider; never require new CORE functions or import a Python implementation just to read calibration metadata.

Scale-bar tick reference extension (Curve Digitalizer 1.2.6)

For SEM-style scale bars with terminal vertical marks, the viewer now defaults to **center-to-center** measured across the two tick centers. This is a selectable assumption, not a universal manufacturer standard. The UI also offers inner and outer references. A candidate must be confirmed before applying physical scale.

The existing public API ID and manifest v1 remain unchanged. In addition to the original scale-bar fields, readers receive:

```
set info [::GX::CurveDigitizerCalibration::read digitized.sem_contour]
if {[dict exists $info scale_bar]} {
    set bar [dict get $info scale_bar]
    set reference [dict get $bar edge_convention]
    # center | inner | outer | unspecified
}
```

For datasets produced before 1.2.6, edge_convention is unspecified, since their geometry cannot safely be inferred. The raw metadata includes curve_digitizer.scale_bar.edge_convention, and the canonical calibration manifest includes scale_bar.edge_convention. No new CORE API or alternate persistence storage is introduced.

Complementary curve-column semantics

Calibration does not assign scientific column roles. For the stable mapping of curve membership, curve label, point order, X, Y and family parameter, use CURVE_DIGITIZER_DATASET_SCHEMA_API_v1 (see CURVE_DATASET_SCHEMA_API_v1.md), also resolved by published Dataset ID.

Additive operations (Curve Digitalizer 1.2.10)

The API ID and namespace remain CURVE_DIGITIZER_CALIBRATION_API_v1; read, hasCalibration and schema preserve their existing arguments and return fields. The new Tcl operations operate on a **published output Dataset ID**, not on the source image ID or an Interactive Document.

```
set identity [::GX::CurveDigitizerCalibration::identity $datasetId]
set physical [::GX::CurveDigitizerCalibration::transformPoints \
    $datasetId {{10.25 20.5} {35 70}} pixel_to_scientific]
set raster [::GX::CurveDigitizerCalibration::transformPoints \
    $datasetId $physical scientific_to_pixel]
set local [::GX::CurveDigitizerCalibration::jacobian \
    $datasetId {{10.25 20.5} {35 70}}]
set errors [::GX::CurveDigitizerCalibration::pointUncertainty \
    $datasetId {{10.25 20.5} {35 70}}]
```

transformPoints datasetId points direction

- points is a Tcl list of {x y} pairs. For pixel_to_scientific, coordinates are source raster pixel centers (0,0) through (width-1,height-1).
- direction is exactly pixel_to_scientific or scientific_to_pixel.
- Results are ordered {x y} pairs in the requested coordinate space.
- The normalized projective homography and each axis's linear/log mapping are handled internally. Inversion uses the homography inverse and inverse axis mappings; no consumer should reimplement this mathematics.
- A singular mapping, nonfinite number, nonpositive log-axis input or point outside the calibrated image/domain raises an explicit Tcl error. For domain violations the error code begins CURVE_DIGITIZER CALIBRATION OUT_OF_DOMAIN; singularity uses CURVE_DIGITIZER CALIBRATION SINGULAR. A batch fails atomically; no partial output is returned.
- The domain is the calibrated original Image Dataset raster. Extrapolation beyond the four calibration corners is deliberately unsupported.

identity datasetId

Returns available=1, calibration_id, sha256 and source_sha256 for 1.2.10+ publications. The ID is cal_ followed by the first 24 hex digits of the calibration SHA-256. The complete fingerprint uses a deterministic UTF-8 canonical JSON encoding (sorted keys, no insignificant spaces) of the image SHA-256, raster dimensions, both scientific axis mappings, the normalized homography and the source-to-calibration pixel mapping. It deliberately **excludes** configuration ID, output Dataset ID, curve names and traced points. Two published curves therefore share the identity when they use the exact same source bytes and geometric calibration, even if digitized independently.

For a pre-1.2.10 publication without fingerprint fields, the method returns available=0, reason=legacy_fingerprint_unavailable. It never fabricates a hash of a historical calibration. The original read fields continue to work.

jacobian datasetId points

Returns a list of dictionaries: valid, point_px, scientific_point, jacobian (2x2 derivative of scientific X,Y with respect to original raster X,Y), and determinant. It handles the local derivative of perspective projection and the actual linear/log scientific axes. Outside-domain or locally singular transformations produce explicit errors instead of an arbitrary matrix.

pointUncertainty datasetId points

Returns one dictionary per input pixel. Without independently supplied uncertainty, each says status=unavailable, reason=no_quantified_uncertainty. There is **no automatic zero-error or perfect-precision assumption**.

If the published metadata explicitly contain a nonnegative uncertainty.pixel_sigma_px, the method reports that measured/declared input sigma and propagates it using the local Jacobian under the **explicit isotropic_independent_pixel_errors** model, returning output_sigma_x and output_sigma_y. Other systematic calibration effects are not automatically accounted for. No uncertainty measurement is invented by the Digitalizer.

Extended read result and calibration metadata

New optional fields are `calibration_id`, `calibration_sha256`, `raster_geometry_json`, `raster_pixel_mapping` (identity for current source pixels, or `legacy_unrecorded`), `calibration_method` and `calibration_method_json`. The stable published metadata additionally include `curve_digitizer.calibration_identity_payload` for independent auditing.

`raster_geometry_json` explicitly records the identity relationship between calibration pixels and the imported Image Dataset's original pixel centers: no in-viewer crop, rotation or resize. It does **not** claim to know any pre-import crop, rotation, resampling or other undocumented upstream image operations (`upstream_processing=unknown`). If a future producer records a nonidentity transform, the reader must support it explicitly; it will not silently treat it as identity. `calibration_method` records the calibration type, point/scale-selection method, tick-edge convention and uncertainty availability without suggesting an unverified metrological accuracy.

The public API uses only the existing Session Inspection and Dataset Metadata services. The core framework sources are unchanged.

Independent repeated calibration measurements

The existing API retains its identifier and all previous default signatures. New publications include `curve_digitizer.calibration_group_id`, `curve_digitizer.active_measurement_id`, `curve_digitizer.measurement_count`, `curve_digitizer.calibration_measurements` (canonical JSON) and indexed `curve_digitizer.measurement.N.*` scalar fields in ordinary Dataset Metadata. They never require an Interactive Document to be present.

```
set trials [::GX::CurveDigitizerCalibration::replicates $datasetId]
set trial [lindex $trials 0]
set id [dict get $trial measurement_id]
set scientific [::GX::CurveDigitizerCalibration::transformPoints \
  $datasetId {{100.0 200.0} {300.0 400.0}} pixel_to_scientific $id]
set identity [::GX::CurveDigitizerCalibration::identity $datasetId $id]
```

Each trial has `measurement_id`, `label`, `state` (draft, accepted, rejected), `active` (0/1), and `valid` (0/1). Valid trials have their own transform, `calibration_id` and `calibration_sha256`. Unknown or invalid trials produce explicit errors when used for transformations. The default call with no ID applies the primary active accepted mapping as before. Legacy datasets without a measurement group report a synthetic single active record (`measurement_id=legacy_active`) without modifying any source record. No sample variability/uncertainty is calculated by the viewer or this API.

Multiple calibration publication layouts

`CurveDigitizerCalibration::replicates datasetId` enumerates **all** calibration measurements known when the Dataset was published, with stable measurement ID, label, state, validity, transform and fingerprint where valid. The Dataset therefore reports other accepted/draft/rejected measurements, not only the one that produced its default X/Y column. A separate publication selects one of these measurements for its scientific numeric rows. A combined publication selects multiple measurements and publishes a `calibration_key` semantic column (see the Dataset Schema API).

The currently selected subset is independently recorded in public metadata:

```

curve_digitizer.publication_calibration_mode = separate | combined
curve_digitizer.publication_calibration_ids = ["meas_...", "meas_..."]
curve_digitizer.publication_measurement_id = meas_... (primary calibration)

```

The default read/identity/transformPoints semantics continue to use the primary calibration stored in the Dataset manifest. Explicitly pass a measurementId to transformPoints (or identity) to use another mapping. Consumers must inspect the publication mode and Dataset Schema roles before assuming there is exactly one calibrated realization per curve.

CURVE DATASET SCHEMA API v1.md

CURVE_DIGITIZER_DATASET_SCHEMA_API_v1

SPDX-License-Identifier: GPL-3.0-or-later

Module: viewer.curve_digitizer

Minimum publication format: curve schema version 1 (Curve Digitalizer 1.2.8)

Goal: retrieve semantic role / Variable ID bindings for scientific curve Datasets using only the published Dataset ID. All public access goes through CORE Session Inspection, Dataset Metadata and Variable Schema v3 APIs.

Why

table.generic and a variable's display label do not encode whether a column is the curve membership key, its human title, or the within-curve point order. Consumers must not assume a particular CSV header, physical column position, Variable ID or UI implementation. Every publication carries a versioned role manifest linked to real CORE Variable Schema IDs.

Stable API

```

# datasetId is the canonical output Dataset ID, current Session is default.
set m [::GX::CurveDigitizerDataset::read $datasetId]
set id [::GX::CurveDigitizerDataset::column $datasetId curve_key]
set supported [::GX::CurveDigitizerDataset::hasSchema $datasetId]
set descriptor [::GX::CurveDigitizerDataset::schema]

```

read datasetId ?session? returns a Tcl dictionary with:

Key	Type	Meaning
api_version	integer	Always 1
curve_schema_version	integer	1; 0 for validated historical fallback
dataset_id	string	Canonical Dataset ID
contract	string	table.generic or curve_family.table

Key	Type	Meaning
mode	string	independent or numerical_family
row_dimension	string	Row Dimension ID of every published column
grouping_role	string	Always curve_key
columns	dict	Semantic role → current Variable ID
variables	dict	Semantic role → CORE V3 Variable Schema descriptor
role_meanings	dict	Stable English meaning for each role
index_origin	integer	Always zero
legacy_inferred	boolean	1 only for validated pre-v1 publications

column datasetId role ?session? returns the mapped Variable ID for a declared role and raises an error when the role is absent. hasSchema returns 1/0; read is the authoritative diagnostic and raises meaningful errors. schema reports the supported contract version, modes, roles and meanings.

The optional session must identify the **current** Session: CORE Variable Schema lookup does not expose a foreign-Session API. The manifest never reads Curve Digitalizer configuration files, private JSON, CSV headers or generated runtime caches.

Roles and precise semantics

Role	Type	Semantics
x	numeric	Calibrated scientific X coordinate of each digitized point, in Variable Schema unit
y	numeric	Calibrated scientific Y coordinate of each digitized point, in Variable Schema unit
curve_key	text	Opaque, stable identifier of the curve to which the row belongs. Group rows by this value; unique within the Dataset
curve_name	text	User-editable display title repeated for each point. Must not be used for grouping, joins or curve identity
point_order	integer	Zero-based digitization order within each curve_key; not a global sequence or an X sort order
family_parameter	numeric	A true numerical family parameter, repeated per point; present only for numerical_family mode

Each row represents **one digitized point**. For independent, columns x, y, curve_key, curve_name, and point_order are mandatory. For numerical_family, family_parameter is additionally mandatory. A numeric family value is **not** a replacement for curve_key: two curves might share one family value and must still be independently addressable.

Scientific axis roles like independent or measured remain user-selectable in Variable Schema; they do not replace this dataset-structure role contract. Plugin Semantic Registry definitions describe curve_series_identifier, curve_series_label, curve_point_index in the V3 variable descriptors.

Canonical Dataset Metadata representation

Every v1 Dataset has the following keys (under the public Dataset Metadata API):

```
curve_digitizer.curve_schema_version = 1
curve_digitizer.curve_mode = independent | numerical_family
curve_digitizer.curve_grouping = curve_key
curve_digitizer.curve_row_dimension = row
curve_digitizer.curve_column.x = <variable_id>
curve_digitizer.curve_column.y = <variable_id>
curve_digitizer.curve_column.curve_key = <variable_id>
curve_digitizer.curve_column.curve_name = <variable_id>
curve_digitizer.curve_column.point_order = <variable_id>
# only for mode=numerical_family:
curve_digitizer.curve_column.family_parameter = <variable_id>
```

The producer version is **not** the schema version. A new producer can change technical Variable IDs, labels and ordering while keeping the same role names and `curve_schema_version=1`. A breaking semantic change requires a new schema version and a suitable backward-compatible API evolution.

All mappings are persisted with the data in CORE Dataset Publication API v2's `structured_metadata` transaction. Renaming a display label does not alter the column mappings. The public API validates the mapped IDs against public Dataset Variable Schema v3, including datatype and row dimension, so it cannot silently accept a nonexistent or wrongly typed column.

Consumer example: two independent SEM contours

```
set manifest [::GX::CurveDigitizerDataset::read sem_contours]
if {[dict get $manifest mode] ne "independent"} {
  error "Not a collection of independent curves"
}
set keyId [dict get $manifest columns curve_key]
set nameId [dict get $manifest columns curve_name]
set orderId [dict get $manifest columns point_order]
set xId [dict get $manifest columns x]
set yId [dict get $manifest columns y]
# Read numeric/text values by these IDs through CORE canonical Dataset APIs.
# Group rows by the opaque value in $keyId, order each group by $orderId,
# then plot each group using the values of $xId and $yId.
```

Do not infer the roles by comparing strings such as `curve_id` or `curve_label`, by matching display labels, or by examining GUI state.

Compatibility and errors

`read` rejects non-Digitizer datasets, invalid schema versions, missing roles, mode/contract mismatch, repeated mapping to the same Variable ID, wrong Variable datatypes, and dimensions inconsistent with the declared row axis.

Pre-1.2.8 publications lack a curve schema manifest. When an output is explicitly identified as Curve-Digitizer-owned and has the known historical technical variable IDs, `read` maps them through a strictly checked historical layout and returns `legacy_inferred=1`, `curve_schema_version=0`. It does not silently guess semantics from a variable's *display label*. Old Dataset files are not rewritten or migrated. The old calibration API is unchanged.

Relation to other public APIs

- `CURVE_DIGITIZER_CALIBRATION_API_v1`: physical/image calibration provenance.
- `DATASET_METADATA_API_v1`: persistent semantic binding fields.
- `DATASET_VARIABLE_SCHEMA_API_v3`: type, unit, Variable ID, semantic role.
- `DATASET_PUBLICATION_API_v2`: atomic data+metadata publication.

Additive source raster semantics (Curve Digitalizer 1.2.10)

The existing CURVE_DIGITIZER_DATASET_SCHEMA_API_v1 remains unchanged in name, return contract and mandatory roles. All new publications add **two numeric Dataset variables** with declared role mappings:

Semantic role	Meaning	Default Variable ID	Unit
raster_x	X coordinate of the original source-image pixel center	pixel_x	px
raster_y	Y coordinate of the original source-image pixel center	pixel_y	px

The variable IDs and human-readable labels are examples, **not** the contract. Consumers must resolve the canonical Variable IDs from the existing columns dictionary returned by `CurveDigitizerDataset::read`. The Variable Schema also contains `curve_raster_x` and `curve_raster_y` roles registered through GX-TXT's existing Semantic Registry API. Scientific X,Y and original raster X,Y coexist in one table, one row per digitized point.

`read` now adds `raster_points_available` (0/1). Existing pre-1.2.10 outputs remain readable and correctly report unavailable raster columns. Partial or invalid X/Y mappings are rejected rather than guessed.

Optional data-centric reader

```
set info [::GX::CurveDigitizerDataset::read $datasetId]
if {[dict get $info raster_points_available]} {
  set batch [::GX::CurveDigitizerDataset::rasterPoints \
    $datasetId "" 0 10000]
  # One filtered batch of the first 10000 dataset rows:
  set left [::GX::CurveDigitizerDataset::rasterPoints \
    $datasetId $curveKey 0 10000]
}
```

`rasterPoints datasetId ?curveKey? ?start? ?count?` reads only the requested bounded row range using **CORE Canonical Dataset API v2**. Default range is `start=0, count=10000`; maximum count is 100000. A nonempty `curveKey` filters just the fetched batch; `start` is the source Dataset row ordinal, **not** a curve-relative offset. The result is a dictionary:

- `dataset_id`, `start`, `consumed_rows`, `coordinate_reference=original_source_image_pixel_centers`;
- `points`: list of dictionaries with `curve_key`, `curve_name`, `point_order`, `x`, `y`, `raster_x`, `raster_y`, `dataset_row`.

Read subsequent row ranges until fewer than `count` rows are consumed. The API never parses CSV files or accesses the Curve Digitizer's private UI state. A historical output without persisted raster columns raises a clear error. The values cannot reliably be reconstructed when the original calibration was lost. Scientific values are not rounded to integer raster positions.

For comparisons of five digitizations of one SEM image, first use the Calibration API identity fingerprints and source SHA-256. Compare the original raster points independently of the transform. Then transform the same positions with `transformPoints` and use `jacobian / pointUncertainty` only where actual uncertainty inputs exist.

Calibration-repeat data independence

`rasterPoints` returns original source-raster points. These remain independent of the active calibration measurement, which allows another module to apply multiple published calibrations to the same pixels through `CurveDigitizerCalibration::transformPoints(datasetId, points, direction, measurementId)`. No additional curve-family semantics or Dataset duplication is implied.

Calibration-aware combined publication

The existing API name and version remain unchanged. In separate publication, `x` and `y` represent points calibrated with a single selected measurement, while `CurveDigitizerCalibration::replicates` still exposes the complete group. Combined publication extends the mapping without changing the mandatory roles:

Role	Current Variable ID example	Semantics
calibration_key	calibration_id	Stable opaque measurement ID; identifies the calibration applied to this row
calibration_name	calibration_label	Editable human-readable calibration label; not a group key

For combined datasets:

```
curve_digitizer.curve_grouping = curve_key_and_calibration_key
curve_digitizer.curve_column.calibration_key = calibration_id
curve_digitizer.curve_column.calibration_name = calibration_label
```

The public read result adds `grouping_roles` (`{curve_key calibration_key}` in combined mode, otherwise `{curve_key}`) and `publication_calibration_mode` (combined or separate). A consumer **MUST** group combined rows by the **pair** (`curve_key`, `calibration_key`) and order points by `point_order` within that pair. `curve_key` alone is insufficient: every original curve has one numeric realization per selected calibration. Legacy/simple consumers should use the separate layout.

For combined output, `rasterPoints` accepts an additional optional final `calibrationKey` filter:

```
set schema [::GX::CurveDigitizerDataset::read $datasetId]
set calColumn [dict get $schema columns calibration_key]
set result [::GX::CurveDigitizerDataset::rasterPoints $datasetId "" 0 10000
$calibrationId]
```

This filter is applied only to the requested CORE row batch, not to the entire Dataset. The original raster positions are intentionally repeated for each calibration so a combined Dataset remains self-contained. The calibration metadata lists **all** available measurements, even when only a subset was selected for a publication. Selection provenance uses `curve_digitizer.publication_calibration_mode`, `curve_digitizer.publication_calibration_ids` (JSON list), and `curve_digitizer.publication_measurement_id` (primary measurement whose calibration is exposed by the default Calibration API read call).

IMAGE PROCESSING API v1.md

Curve Digitalizer Image Processing API

SPDX-License-Identifier: GPL-3.0-or-later

Public API ID: CURVE_DIGITIZER_IMAGE_PROCESSING_API_v1

Provider: viewer.curve_digitizer (GX-TXT CORE plugin)

Tcl namespace: ::GX::CurveDigitizerImage

Script Bridge provider: plugins/modules/viewer.curve_digitizer/api/script_bridge_binding.py

Purpose

Read the history and provenance of ImageMagick photometric enhancements that were published as canonical image.raster Datasets. Consumers need only a canonical Dataset ID, not the implementation details of Dataset Metadata API sidecars. The source image and its SHA-256, processing sequence and original-coordinate identity remain accessible independently of the Interactive Document.

This API does not process images, write metadata, make files or modify scientific curves. It is a stable, read-only interpretation boundary. The normal CORE DATASET_FILE_PUBLICATION_API_v1 owns output publication and SHA-256.

Image processing and geometry guarantees

Operations are deliberately limited to these photometric filters: grayscale, channel, gamma, brightness_contrast, auto_level, contrast_stretch, clahe, threshold, median, negate, sharpen. Image dimensions are checked against the original after processing, and publication verifies the PNG header dimensions against structured metadata. Crop, resize, rotate, flip, shear, perspective and arbitrary command execution are not supported. pixel_transform=identity and geometry_preserved=true mean the **raster coordinate system and grid** have not changed; filters can change the apparent placement/contrast of an edge and do not provide new geometric metrology. Published originals are never overwritten.

TCL API

```
set manifest [::GX::CurveDigitizerImage::read $datasetId]
set operations [::GX::CurveDigitizerImage::operations $datasetId]
set lineage [::GX::CurveDigitizerImage::lineage $datasetId]
set supported [::GX::CurveDigitizerImage::hasManifest $datasetId]
```

Optional second argument session must equal the currently open GX-TXT Session. An unknown or unrelated Dataset yields an explicit error from read; hasManifest returns false. lineage recursively validates parents, bounded to 64 links, and rejects missing/mismatched parent SHA-256 or cycles. It includes the original Image Dataset as the last element. No image bytes are copied.

read **return fields**

Field	Contract
api_id, api_version	Stable public API identity and major version
dataset_id	Canonical output Dataset ID
source_dataset_id	Immediate Image Dataset parent
source_display_name	Human-readable source name at publication
source_sha256	SHA-256 of immediate canonical parent
processing_operations	Ordered list of dicts {kind, value}
operation_count	Number of ordered steps
processing_engine	ImageMagick
processing_engine_version	Captured installed runtime version
geometry_preserved	1 / true, validated
pixel_transform	identity
image_width, image_height	Output raster dimensions in source pixels
processing_fingerprint	Deterministic SHA-256 of source identity, recipe and runtime version
output_sha256	CORE-registered output payload SHA-256

A separate human description is stored with the published Dataset notes. The canonical Dataset ID remains the stable image identity.

A processing fingerprint identifies a specific recipe/source/runtime tuple. It does not establish numerical equivalence between different engine releases. `source_sha256` and `output_sha256` identify the exact image bytes.

Errors

Missing Dataset, incompatible contract, missing/corrupted manifest, unsupported schema, invalid dimensions, unknown operations and broken lineage raise Tcl errors (or an explicit Script Bridge RemoteError). Missing fields are never filled by guessing. Legacy Image Datasets without a processing manifest are still ordinary valid `image.raster` sources, but `read` does not falsely claim they were processed.

Python Script Bridge

```
from gtxtxt import gtxtxt_api_require, gtxtxt_api_call

gtxtxt_api_require('CURVE_DIGITIZER_IMAGE_PROCESSING_API_v1@1:viewer.curve_digitizer')
image_id = 'sem.enhanced'
info = gtxtxt_api_call('CURVE_DIGITIZER_IMAGE_PROCESSING_API_v1', 'read', [image_id])
for op in info['processing_operations']:
    print(op['kind'], op['value'])
print('Original:', info['source_dataset_id'], info['source_sha256'])
parents = gtxtxt_api_call('CURVE_DIGITIZER_IMAGE_PROCESSING_API_v1', 'lineage',
[image_id])
```

GNU Octave Script Bridge

```
assert(gtxtxt_api_has('CURVE_DIGITIZER_IMAGE_PROCESSING_API_v1'));
info = gtxtxt_api_call('CURVE_DIGITIZER_IMAGE_PROCESSING_API_v1', 'read',
{'sem.enhanced'});
disp(info.source_dataset_id);
for k = 1:numel(info.processing_operations)
    disp(info.processing_operations{k});
endfor
```

Both clients use the common provider-owned Script Bridge. The module registers its `api_id`, version, argument/result descriptors and callable operations; Bridge discovery verifies the exact `module.ini` declaration. Scripts have no access to the Tcl implementation's storage paths or private viewer state.

Persistence and source ownership

All fields are written as strings using `DATASET_METADATA_API_v1` inside the file-publication transaction. The implementation uses a namespaced metadata layout currently beginning `curve_digitizer.image_processing.` and indexed `operation.N.kind/operation.N.value` keys. This is **not** a consumer contract: callers must use `read/operations/lineage`. Changes to internal storage may be implemented later without altering public operation semantics.

Security and compatibility

ImageMagick is selected from CORE's `runtime.magick` preference via `runtimeToolStatus magick`. Only allowlisted command arguments are accepted; no shell or user-entered ImageMagick expression is executed. A missing runtime disables enhancement/publication without disabling curve digitization. The output PNG is staged into a host-created temporary file, checked and published through `datasetFilePublicationPublish`. The stage is deleted after success or error. CORE assigns the producer namespace and prevents the viewer from overwriting an unrelated Dataset.