

Description

Pixometry is an Enfocus Switch app that connects a Switch flow to **Pixometry**, a cloud image-processing service used by print shops, pre-press operations, and photo studios worldwide. Drop the Pixometry flow element into any flow and every image that arrives on its incoming connection is sent to Pixometry's cloud engine for automated processing.

Typical uses:

- **Image enhancement** — sharpening, noise reduction, tone, colour balance, skin-tone correction, blue sky and vegetation adjustments, selective and brand-colour corrections.
- **Background removal** — AI-driven cutouts producing transparent PSD, TIFF, or PNG output.
- **Smart AI upscaling** — low-resolution sources brought up to print resolution via Topaz Gigapixel.
- **Colour-managed pipeline** — sRGB / AdobeRGB / eciRGB / CMYK / Grayscale, with ICC-profile embedding and conversion.
- **PDF processing** — every image inside a PDF processed individually and re-embedded, preserving the original PDF structure.

Processing behaviour is defined once in the Pixometry webapp as a *channel* (a named preset of settings). The Switch element then selects the appropriate channel per job. Channel selection, crop rectangle, and target output size can all be driven by Switch variables so a single element can serve many workflows.

Each processed job produces the finished image on the success output **plus a companion XML log** that lists the source file, the chosen channel, the crop and size parameters, the processing steps the engine performed, and the output filename. The log is routed alongside the image (or alongside the original on failure) so downstream elements can archive, parse, or ignore it.

Compatibility

Enfocus Switch 22 update 1 and higher. The app uses the Node.js scripting interface introduced in Switch 20; it has been tested on the current Switch release on both macOS and Windows.

Compatibility third-party applications

The app communicates with the **Pixometry cloud service** over HTTPS. Pixometry is a hosted SaaS platform — there is no version to pin against, and no software to install alongside Switch. A user account (with credits) is required.

- **Product page:** <https://pixometry.com>
- **Sign up / free trial:** <https://app.pixometry.com/#/register> (50 credits included)
- **API keys:** created inside the webapp on the *API* page after login

- **Help page:** <https://app.pixometry.com/#/help/switch>

Application discovery details

There is nothing to discover on the local system. The app talks to Pixometry entirely over HTTPS using two element properties:

1. **API base URL** — the Pixometry endpoint. Default `https://app.pixometry.com`. Only change this to `https://staging.pixometry.com` if the operator is running an internal test against the staging environment.
2. **API key** — a personal token from the operator's Pixometry account. Generated at `app.pixometry.com` → API → New API key. The key is shown once at creation, copy it into the Switch element's *API key* property. Keys start with `pk_live_` (production) or `pk_test_` (test-mode; not needed for normal use).

A valid *API key* pointing at the right *API base URL* is enough to reach the service. The element verifies connectivity at design time by fetching the tenant's channel list (used to populate the *Channel* property's library picker); a bad key or unreachable server surfaces as a design-time error in Switch Designer.

Connections

The Pixometry element uses Switch's standard **TrafficLight** connection layout:

Incoming

- **One incoming data connection, required.** Accepts individual image files (JPG, PNG, TIFF, PSD, HEIC, WebP, AVIF, and other formats supported by the Pixometry engine) or PDF documents. Folders are not supported — configure the upstream element (e.g. *Split multi-file job*) to feed one file at a time.

Outgoing

- **Data — Success:** the processed file. Renamed by the element to the filename returned by the engine so extension changes (for example TIFF → PSD when a channel adds a cutout) propagate correctly.
- **Data — Error:** the original input file, unmodified, when processing failed (invalid API key, insufficient credits, engine error, unsupported input, timeout, etc.).
- **Log — Success:** a companion XML log describing the successful job (source, channel, crop, size, processing steps, output filename, timings). Routed alongside the processed file so success-side archivers see both.
- **Log — Error:** an XML log describing the failed job (source, channel, failure reason, timestamps). Routed alongside the original file on the error side.

The XML log filename is always `<datafile>.xml` — so a processed `photo.jpg` is accompanied by `photo.jpg.xml`. The log root element is `<pixometry-log version="1">`. All fields are optional; missing values are omitted from the output. The API key is never included in the log.

Properties detailed info

The Pixometry element exposes 12 properties. Only *API key* and *Channel* are strictly required to process; everything else is optional and driven by the channel's own configuration in the webapp unless the operator wants to override on a per-job basis via Switch variables.

Every property with a text editor supports Switch variables, so values can be bound to job metadata, dataset XPath expressions, private-data fields, or JobTicket fields — allowing one element to drive many different processing outcomes based on incoming data.

Flow elements properties

Property	Description
API base URL ApiBase	The Pixometry endpoint the element talks to. Default <code>https://app.pixometry.com</code> . Change only when testing against a non-production environment. Supports Switch variables.
API key ApiKey	The API key generated in the Pixometry webapp under <i>API</i> . Masked in Switch Designer and redacted in flow exports. Required.
Channel Channel	The channel to process against. Accepts either the numeric channel ID or the channel name (channel names are unique per tenant). Three editors available: <i>Select from library</i> (dropdown populated from the tenant's channels at design time), <i>Single-line text with variables</i> (bind to job metadata for per-job channel selection), and <i>Script expression</i> (JS expression for advanced routing rules). Required.
Job tags (JSON) TagsJson	Optional flat JSON object of string-to-string pairs, e.g. <code>{"MsgForInspector":"CHP_NWS_COL"}</code> . Tags are forwarded to Pixometry and surfaced to channel variants and conditions, allowing a single channel to route work based on job metadata. Leave empty if the target channel does not use tag routing. Supports Switch variables so the JSON can be assembled from job metadata at run time.
Crop X (pixels) CropX	Optional. Top-left X coordinate of the crop rectangle in source pixels. Common binding for sidecar-driven flows: <code>\${[Dataset.XML:/photo/crop/x]}</code> . Set 0 for a crop that starts at the left edge; the element treats an empty value as "no crop coordinate given" and a numeric 0 as "start at pixel 0".
Crop Y (pixels) CropY	Optional. Top-left Y coordinate of the crop rectangle in source pixels. Same semantics as <i>Crop X</i> .
Crop width (pixels) CropWidth	Optional. Width of the crop rectangle in source pixels. Must be > 0 to count as "set"; an empty value or 0 means "do not crop".
Crop height (pixels) CropHeight	Optional. Height of the crop rectangle in source pixels. Must be > 0 to count as "set".
Output width (mm) OutputWidthMm	Optional. Target output width in millimetres. Converted to pixels at <i>Output DPI</i> . If both <i>Output width</i> and <i>Output height</i> are set, fit-in-box resizing is applied. If only one dimension is set, the other is derived from the crop aspect ratio.
Output height (mm) OutputHeightMm	Optional. Target output height in millimetres. See <i>Output width</i> for combined behaviour.
Output DPI OutputDpi	Optional. Target output DPI for resampling. Required if either output-dimension property is set. When used stand-alone (no width/height), it stamps the output DPI metadata without resizing pixels.

Property	Description
Job timeout (minutes) JobTimeoutMinutes	Optional. How long the element waits for the server to finish a single job before failing it to the Error output. Default 30 minutes (adequate for images). PDFs with many pages or embedded images may need 120+ minutes.

Per-job crop and size overrides merge with the channel's own Size settings, they do not replace them. If a channel has a fixed output size baked into its *Size* panel in the Pixometry webapp, every submission to that channel will be resized to that size regardless of the operator's overrides. To grant an element full control over crop and size, use a channel whose *Size* panel is not configured (or explicitly set all three *Output width*, *Output height*, and *Output DPI* properties to override).

Outgoing connections properties

The Pixometry element uses Switch's standard TrafficLight connection layout: no per-connection properties are exposed on any of the four outgoing connections. Route them by their Data-success / Data-error / Log-success / Log-error semantics.

Publisher

Elpical Software B.V.
Trading as Pixometry
The Netherlands
pixometry.com

Support

support@pixometry.com
Help & docs: app.pixometry.com/help/switch
Sign up / free trial: app.pixometry.com/register