



Paperweight Manual

User and administrator guide

Document capture, configuration,
deployment, and recovery.

Product 0.1.0

Documentation revision 2

2026-09-12 · Public preview

Clinimatix, LLC

paperweight.clinimatix.com

Table of Contents

Get started

Overview and requirements

Install and configure

3

5

Using Paperweight

Scanner capture

Folder intake

Processing and activity

8

10

12

Configuration

Configuration overview

Output profiles

Sources

Destinations

Limits and retention

14

16

19

21

23

Deployment

Deployment options

Rollout and maintenance

25

27

Operations and recovery

29

Troubleshooting

31

Reference

Commands and scripts

Release notes and limits

Glossary

Third-party notices

33

35

36

37

Paperweight user and administrator guide

Paperweight helps your team turn scans and incoming documents into consistent PDF or TIFF files. Capture pages from a scanner or use a watched folder, then let Paperweight apply your organization's settings, check the result, and deliver it to the selected destination folder.

Product 0.1.0 · Documentation revision 2 · 2026-09-12 · Public preview

Start here

- **Operators:** follow [Scanner capture](#), then learn to read [processing activity](#).
- **Administrators:** start with [Install and configure](#), review the [configuration reference](#), and choose a [deployment strategy](#).
- **Support staff:** use [Operations and recovery](#) and [Troubleshooting](#).

Prefer an offline reference? Download the **Paperweight Manual** to read this guide as a PDF.

Requirements

Requirement	What to prepare
Endpoint	Windows 11 x64; a standard-user interactive session for operation
Installation	Administrator or SYSTEM access to install the per-machine MSI and deploy policy
Scanner	Separately installed vendor TWAIN or WIA driver for your scanner model
Working storage	Protected fixed local disk for each user's job store; no UNC spool or known OneDrive location
Destination	Existing local or SMB intake folder and a separate staging folder on the same filesystem root
Access	The signed-in operator needs the actual scanner, source, staging, and destination permissions

The application includes its .NET runtime, so operators do not need to install it separately.

What you can do

Capture and review scanned pages, process incoming files, and track local activity and storage savings. Paperweight keeps each job's originals and settings for recovery, checks the processed document, and delivers files without overwriting existing documents. Use the Capture, Activity, Configuration, and About tabs to get around. Closing the window leaves Paperweight running in the notification area.

Folder intake considers PDF, TIFF, JPEG, and PNG files. The actual format is checked from content, not trusted from the extension. Profiles produce PDF or TIFF. There is no OCR, semantic classification, Windows service, EHR API, management server, or telemetry upload in this version.

⊗ **IMPORTANT**

This guide covers the current Paperweight preview. See [Release notes and limits](#) for features and known limitations.

What Published means

Published means Paperweight completed a filesystem handoff. It does not mean a document was accepted by an EHR or another downstream application. An absent destination file after interruption can mean that a consumer already removed it. Reconcile uncertain delivery before taking action.

Install and configure

Use these steps to install Paperweight and set up your first scanning or folder workflow. Your Clinimatix contact can provide the deployment kit.

1. Prepare the endpoint

Install the vendor's driver for your scanner model separately. Create distinct input, destination, and staging folders. The destination must already exist; staging must be on the same filesystem root and excluded from downstream ingestion. Verify access as the actual signed-in operator, even if installation runs as SYSTEM.

Paperweight stores working files and recovery copies in `%LOCALAPPDATA%\Clinimatix\Paperweight` by default. Provide enough disk space and apply your organization's storage protection and retention requirements. See [Limits and retention](#).

2. Install the application

Use the deployment wrapper with the signed MSI from your deployment kit. See [Deployment options](#) for commands and return codes. Installation is per machine. The default MSI property `AUTOSTART=1` starts the application in the tray at the next user logon; operators scan from their own Windows sessions.

3. Prepare policy

Start from the [complete disabled example](#). Replace the example workstation, Windows account, and directories. Assign each source to one workstation and user, then select its profile and destination. Keep sources disabled until their folders and settings are ready.

```
{
  "schemaVersion": 1,
  "revision": 1,
  "profiles": [
    { "id": "conservative-pdf", "revision": 1, "format": "Pdf", "color": "Preserve", "allowLossy":
false }
  ],
  "destinations": [
    { "id": "document-intake", "folder": "C:\\PaperweightLab\\Intake", "stagingFolder":
"C:\\PaperweightLab\\Staging", "naming": "PreserveStem" }
  ],
  "sources": [
    { "id": "scanner-capture", "acquisition": "Scanner", "profileId": "conservative-pdf",
"destinationId": "document-intake", "ownerMachine": "EXAMPLE-PC", "ownerUser": "EXAMPLE\\Operator",
"enabled": false },
    { "id": "folder-input", "acquisition": "Folder", "folder": "C:\\PaperweightLab\\Input",
"completion": "StableFile", "stableSeconds": 10, "profileId": "conservative-pdf", "destinationId":
"document-intake", "ownerMachine": "EXAMPLE-PC", "ownerUser": "EXAMPLE\\Operator", "enabled": false }
  ],
  "clinicalMode": false,
  "protectedStorageConfirmed": false
}
```

This example uses conservative PDF settings. The [configuration reference](#) explains every setting, including fields omitted here that use defaults. Replace the sample paths with the folders your workflow will use.

4. Validate and deploy policy

Run configuration validation using the installed executable:

```
& 'C:\Program Files\Clinimatix Paperweight\Paperweight.Desktop.exe' `
--validate-config 'C:\Deployment\configuration.json'
```

Exit 0 means the settings pass Paperweight's configuration checks; exit 2 means a setting needs correction. This command checks the configuration file; device connections and folder access are checked when you use the workflow.

Deploy from an administrator session using the deployment kit:

```
.\Deploy-Paperweight.ps1 -Action Configure `
-ConfigurationPath .\configuration.json
```

The wrapper validates the new file, protects the machine policy directory, and preserves the previous policy as `configuration.previous.json`. The managed policy location is `%ProgramData%\Clinimatix\Paperweight\configuration.json`.

5. Capture your first document

Set `enabled: true` on the source you want to use and redeploy policy. Open Paperweight as its assigned user and look for the Ready state. For a scanner source, scan your pages, review the batch, and choose **Finish and process**. For a folder source, save a completed document to the input folder. Follow the result in **Activity**, then open the document in its destination application.

NOTE

Missing policy or no enabled sources assigned to the current machine/account leaves intake inactive. A running application retains its last valid policy if an update is rejected; a fresh launch with invalid policy has no valid workflow to activate.

Scanner capture

Scan directly in Paperweight using a supported TWAIN or WIA device. Select the TWAIN or WIA provider supported by your scanner driver. Paperweight does not automatically switch providers or repeat a scan.

Capture a batch

1. Open **Capture** and choose an assigned scanner workflow. Its output profile and destination come from managed configuration.
2. Select **Find scanners**, then choose the device/provider pair.
3. Choose **Source**, **Color**, **DPI**, and **Page size** from the reported capabilities. Select **Use scanner's settings dialog** if the driver requires it or your administrator's setup instructions call for it.
4. Choose **Scan / add pages**. Acquired pages remain local until the batch is finished. Add another group if needed.
5. Review thumbnails and page order. Confirm all required pages are present and legible.
6. Choose **Finish and process**. The completed batch is queued for processing; follow its progress in **Activity**.

Scanner settings

Control	Behavior
Source	Flatbed, Feeder, or Duplex when supported by the selected provider
Color	Color, Grayscale, or BlackAndWhite when reported by the driver
DPI	Device-reported values; capture requests are constrained to 50-1200 DPI
Page size	Letter, A4, or Legal, subject to reported scan-area support
Native dialog	Opens the vendor's scanner controls; required when the driver reports incomplete capability information

The UI prefers 300 DPI and Color when available. Scanner settings are distinct from the output profile: scanning in grayscale already affects the acquired master, regardless of a later profile's `color`: `Preserve`. Choose capture and output settings together to preserve the detail your team needs.

Paperweight saves scanner preferences locally. It uses lossless PNG capture masters and does not request automatic deskew, blank-page removal, cropping, or stretching. Check the scanner driver's own settings for features that could change or remove page content.

Incomplete or interrupted scans

Cancel scan, an empty feeder, a jam, or a disconnect can leave an incomplete batch. Paperweight retains acquired material instead of silently declaring it complete.

- Add missing pages where possible, then choose **Resolve scan error** and explicitly confirm completeness and order before finishing.
- **Start another batch** leaves the unfinished batch Held; it does not erase the acquired pages.
- After a restart, select a retained incomplete scanner batch in **Activity** and use **Resume incomplete capture**. This is available for captures that have not yet been submitted for processing.

⊗ **CAUTION**

Review the captured pages even when the scan finishes without an error. Check fronts/back, faint text, handwriting, highlights, and orientation.

Closing the application window hides it to the tray. Use the tray's **Exit** command to end the session. Folder processing requires the application and user session to remain running.

Folder intake

Keep using your existing scanner or capture application by having it save completed documents to a folder that Paperweight watches. An enabled folder source processes supported files directly in that folder, without searching subfolders. Supported filename extensions are `.pdf`, `.tif`, `.tiff`, `.jpg`, `.jpeg`, and `.png`; Paperweight also checks the file's contents to confirm its format.

Choose a completion contract

Mode	Capture application behavior	Setup guidance
<code>AtomicHandoff</code>	Write to a temporary filename, close it, then rename into the watched directory with its final supported extension	Verify the producer never writes directly to an eligible final filename
<code>ReadyMarker</code>	Finish and close the document, then create <code><document-name>.ready</code> containing that document's 64-character SHA-256 hex digest	Verify marker timing and hashes; a stale marker cannot authorize changed bytes
<code>StableFile</code>	Leave size and modification time unchanged for <code>stableSeconds</code> , with write-excluding access possible	Test slow saves and network delays: a pause in writing does not prove that a file is complete

`StableFile` defaults to ten seconds. For a new producer integration, prefer a demonstrable completion contract. A temporary extension must not be one of the eligible input extensions.

Ready-marker example

After the producer has closed this synthetic file:

```
$document = 'C:\PaperweightLab\Input\sample.pdf'
$digest = (Get-FileHash -LiteralPath $document -Algorithm SHA256).Hash
Set-Content -LiteralPath ($document + '.ready') -Value $digest -NoNewline
```

Do not use an empty sidecar. The marker records content identity, not only permission to scan a pathname.

Polling and source health

Paperweight checks folders every two seconds and can also respond to file-change notifications. Processing a document can delay the next check. If a folder is unavailable, Paperweight gradually increases the time between attempts, up to sixty seconds.

The **Configuration** tab shows source health. `Ready` means the source was successfully enumerated, not that every downstream document was accepted. Other states can identify unavailable sources, reconciliation needs, access failures, or a file-specific rejection.

Originals and duplicate notifications

Paperweight takes a protected local snapshot and records file identity and content hash before processing. Repeated notifications are not separate submissions of the same acquired file. New or replaced content is treated separately.

After publication, source cleanup verifies the original file's identity and content. A replacement at the same pathname is preserved. The local job directory then serves as the recovery archive, subject to [retention](#).

⊗ IMPORTANT

Assign each source to one workstation and Windows account. Do not enable competing consumers or point multiple sources at overlapping directories. Choose a completion mode that matches how your capture application finishes saving files.

Processing and activity

Whether you scan pages or use a watched folder, Paperweight keeps protected copies of the originals and the settings for that job. It checks the input, applies the selected output profile, checks the result, and delivers the file to its destination.

Normalization and validation

Paperweight preserves grayscale and black-and-white values where possible. TIFF output uses Group 4 compression for wholly black-and-white batches and ZIP otherwise. When rebuilding PDF pages as images is unsuitable, Paperweight can optimize the existing PDF structure. Already-efficient files may pass through unchanged if they meet your selected settings.

Output is reopened and checked for format, byte limits, page count, dimensions, codecs, and supported restrictions. Appearance/pixel comparisons guard transformations. Have the staff who use the documents compare sample output with the originals for readability and completeness; automated checks cannot judge which details matter to their work.

Read Activity

Activity exposes job status, error code, original/result bytes, page count, and timing. Stored operational details also include acquisition, profile/revision, destination ID, configuration revision, processing decision, and a stable installation ID. The visible table is local to the signed-in user's job store.

State	Meaning
Acquiring	Originals are being captured or snapshotted
Queued / Processing	Ready for the worker / worker running
Validated	Output passed checks; publication remains
Publishing	Publication intent recorded; filesystem handoff in progress
Published	Final file handoff recorded; source cleanup/archive may remain
Archived	Successful cleanup completed; local recovery material retained
Held	Review required before further action
Uncertain	Delivery cannot be established safely; downstream reconciliation required
Expired	Eligible recorded archive artifacts were removed

Measure efficiency carefully

Per-job reduction is $100 \times (1 - \text{resultBytes} / \text{originalBytes})$ when original bytes are positive. Negative values mean output grew. Compare similar input types and profiles; scanner PNG masters and direct folder imports have different baselines. No fixed savings percentage applies to every document.

The goal is to fit the same document workload into less disk space, without reducing how much your team captures. A preliminary trial in a clinical environment measured up to 90% less disk space consumed by filed documents normalized with Paperweight automations compared with documents captured directly with Canon scanning software. Savings vary by document and settings; review output quality as well as file size.

Team reporting

Activity and efficiency figures are currently available on each workstation. Cross-team reporting and trends are planned; this version does not include a fleet dashboard, reporting API, or automated trend exports.

Operational history excludes source filenames/paths and document content, but can include Windows account identities associated with capture and processing. File ownership or inferred writer information does not establish who scanned a document. Protect this account information according to your organization's policies. Recovery records also contain paths needed for cleanup; do not upload raw job data to public tools.

Configuration overview

Paperweight uses one administrator-managed JSON policy. It links sources to output profiles and destinations by ID. The Configuration tab displays policy/source information and provides **Reload configuration**; it is not a policy editor.

Policy fields

The **Default** column shows what happens when you omit an optional setting. Include every field marked required; explicit `null` is valid only where documented. Unknown setting names are rejected. Use the spelling and capitalization shown in the tables.

Option	Default	Accepted values	Behavior
<code>schemaVersion</code>	required	1	Required configuration schema; other versions fail
<code>profiles</code>	required	Array of output profiles	Policies available to source references; do not use null
<code>destinations</code>	required	Array of destinations	Publication and staging paths; do not use null
<code>sources</code>	required	Array of sources	Workstation/user workflow assignments; do not use null
<code>limits</code>	null	Limits object or null	Null/omitted uses the default resource limits
<code>archiveDays</code>	7	7 only	Fixed successful-archive retention policy
<code>historyDays</code>	90	90 only	Fixed activity-history retention policy
<code>clinicalMode</code>	false	true, false	Requires <code>protectedStorageConfirmed: true</code> and <code>validatedForClinicalUse: true</code> on every profile
<code>protectedStorageConfirmed</code>	false	true, false	Records the administrator's confirmation that storage meets the organization's protection requirements
<code>revision</code>	1	Integer >= 1	Revision included in each job's configuration snapshot

IDs and references

Profile, destination, and source IDs must contain 1-64 ASCII letters, digits, underscores, or hyphens. IDs are unique within each collection, ignoring case. Source references must match the profile/destination ID exactly, including case. Use stable IDs and increment revisions when behavior changes.

All configured paths must be absolute and non-overlapping, including the local data root. Existing reparse points are rejected. Extended device paths and alternate data stream paths are not accepted. Keep input, staging, final destination, and local job data in separate directory trees.

Changes and snapshots

The app checks for policy file changes while running and reloads at an idle boundary. An active capture or operation delays reload. **Reload configuration** can also request an explicit refresh. Invalid replacement policy leaves the running process's previous valid configuration active; it does not silently apply some fields.

Each job keeps a copy of its settings and profile revision. Updating machine policy does not change queued or held jobs. Retrying a job uses its original settings. Let operators know when settings will change and record the revision deployed.

Clinical mode

`clinicalMode: true` requires two administrator confirmations: working storage meets your organization's protection requirements, and each output profile preserves the information your staff need from their clinical documents.

Record those confirmations with `protectedStorageConfirmed: true` and `validatedForClinicalUse: true` on **every** configured profile. These values record your team's review of storage and document quality. Paperweight checks that they are set; it does not assess storage protection or clinical readability itself.

See [Output profiles](#), [Sources](#), [Destinations](#), and [Limits and retention](#). Start from the [complete example](#), with workflows disabled until ready.

Output profiles

Profiles define the output format and transformations a workflow may use. Changes require an administrator-managed configuration update.

Start conservatively

Start with color preserved and lossy processing disabled. Before enabling settings that reduce image detail, have staff compare sample output with the originals for readability and completeness.

```
{
  "id": "generic-conservative-pdf",
  "revision": 1,
  "format": "Pdf",
  "color": "Preserve",
  "allowLossy": false
}
```

This is a profile fragment, not a complete configuration. See the [complete workstation example](#).

Core options

Option	Default	Accepted values	Behavior
id	required	1-64 letters, digits, _ or -	Stable profile identifier
revision	1	Integer >= 1	Version of this profile's behavior
format	Pdf	Pdf, Tiff	Output file format
color	Preserve	Preserve, Grayscale, Monochrome	Pixel color policy
allowLossy	false	true, false	Permits lossy pixel transformations and eligible JPEG encoding
maxDpi	null	1-1200 or null	Maximum output resolution, not an upsampling target
jpegQuality	90	1-100	JPEG quality when eligible PDF raster output uses JPEG
monochromeThreshold	128	Byte 0-255	Threshold for explicit monochrome conversion
allowPdfFlattening	false	true, false	Permits rendering complex PDF pages into raster output

Option	Default	Accepted values	Behavior
<code>allowSignedPassthrough</code>	false	true, false	Allows eligible signature-bearing PDFs only as unchanged bytes
<code>maxOutputBytes</code>	null	Positive integer bytes or null	Hold output exceeding this size; no guaranteed compression target
<code>maxPdfVersion</code>	17	14-17	Maximum PDF version: 14 = 1.4 through 17 = 1.7
<code>metadata</code>	Preserve	Preserve, RemoveDescriptive	Preserve metadata or explicitly allow descriptive metadata removal
<code>rotation</code>	0	0 only	Nonzero rotation is currently unsupported
<code>maxWidthInches</code>	null	Positive number or null	Maximum permitted page width
<code>maxHeightInches</code>	null	Positive number or null	Maximum permitted page height
<code>allowedCodecs</code>	null	Array or null	Optional allowed codec names; see below
<code>validatedForClinicalUse</code>	false	true, false	Records your team's review that this profile preserves the information needed for its intended clinical documents

Before enabling lossy processing

⊗ IMPORTANT

Reducing resolution (`maxDpi`) or changing color policy requires `allowLossy`: `true`. Review representative output before clinical use. Automated image checks cannot replace a review of clinical readability and completeness.

With lossy processing enabled, eligible raster pages above eight bits per pixel can use JPEG when exported to PDF. TIFF output uses Group 4 for entirely bilevel batches and ZIP otherwise; `jpegQuality` is not a TIFF JPEG selector. A preserved grayscale master does not recover color already lost at capture.

`allowPdfFlattening` is a separate permission. Even without downsampling, flattening changes PDF structure and can lose interactive or semantic information. Enable it only when your workflow needs it and your team has checked that the finished documents retain the content and features they need.

Codec and geometry restrictions

Allowed codec names are `Flate`, `DCT`, `CCITT`, `LZW`, `Uncompressed`, `JPX`, and `JBIG2`. Listing a codec permits it; it does not select that compression method for output. An empty array allows no codec, while null means no codec restriction. Paperweight holds complex PDFs when it cannot confirm that they meet these restrictions.

Width/height and byte limits are validation ceilings. They do not request cropping, page resizing, or repeated compression until a target is met. `maxPdfVersion` is also a validation ceiling, not a promise to convert every PDF to an earlier version.

Signatures and metadata

Signature-bearing input is held unless unchanged PDF passthrough is explicitly allowed and other constraints permit it. A passthrough requires PDF output, preserved metadata, and no requested pixel change. The original bytes remain unchanged; Paperweight does not verify cryptographic signature validity.

Existing PDFs with descriptive metadata use a structure-preserving path when necessary. Conversions or flattening that cannot preserve that metadata are held unless removal was explicitly permitted. Embedded raster color profiles and unsupported complex transformations are held conservatively. Active or encrypted PDFs are held even when other permissions are enabled.

Sources

A source selects an acquisition method, output profile, destination, and owning workstation/account. Only enabled sources owned by the current Windows machine and user run.

Option	Default	Accepted values	Behavior
<code>id</code>	required	1-64 letters, digits, _ or -	Stable source identifier
<code>acquisition</code>	required	Folder, Scanner	Source acquisition method
<code>profileId</code>	required	Existing profile ID	Exact case-sensitive profile reference
<code>destinationId</code>	required	Existing destination ID	Exact case-sensitive destination reference
<code>folder</code>	null	Absolute path or null	Required for Folder; not used for Scanner
<code>completion</code>	StableFile	AtomicHandoff, ReadyMarker, StableFile	Folder producer-completion contract
<code>stableSeconds</code>	10	Integer >= 1	Quiet period for StableFile; validated even for other modes
<code>ownerMachine</code>	null	Machine name or null	Required when enabled; compared ignoring case
<code>ownerUser</code>	null	Windows account name or null	Required when enabled; compared ignoring case
<code>enabled</code>	false	true, false	Opt-in to workflow execution
<code>readLocalAudit</code>	false	true, false	For folder sources, attempt to identify a local file writer from existing Windows Security audit records; requires permission to read the log

Workstation and user ownership

Use the workstation's Windows machine name and the operator's resolved identity, for example `EXAMPLE-PC` and `EXAMPLE\Operator`. Confirm the values in the actual account's session. Installation as SYSTEM does not transfer SYSTEM's share permissions to the operator.

A shared fleet deployment can include multiple distinct source assignments. Every path still has to pass policy validation on the endpoint, and roots may not overlap. Prefer narrowly scoped per-workstation policy where a large common policy would introduce irrelevant paths or ownership mistakes.

Scanner and folder differences

Scanner sources appear as selectable workflows in Capture. Device selection and source/DPI/color/page-size controls are user-local scanner preferences, not fields in this source policy. Managed profiles standardize the output policy; this version does not centrally lock every vendor scanner setting.

Folder sources are polled only at the top level. `completion` and `stableSeconds` are meaningful for folder processing; they do not control scanning. See [Folder intake](#) for producer responsibilities and ready-marker contents.

Optional file-writer information

`readLocalAudit` is off by default. When enabled for a folder source, Paperweight looks for recent matching file-write events in the workstation's Windows Security log. It does not enable auditing or grant log access. Your administrator must already have configured suitable audit coverage and access.

A match is evidence of an account that wrote the file, not proof of who scanned or originally created it. Missing, inaccessible, or conflicting records leave the writer unknown and do not stop document processing. Network shares require evidence from the file server; this preview does not collect it. Keep this setting disabled unless your organization needs the information and has approved the required access.

NOTE

Changing ownership or enabling a source does not migrate another user's retained jobs. Preserve local state and coordinate the change to avoid competing consumers or duplicate downstream delivery.

Destinations

A destination defines a final intake directory and a separate staging directory. Paperweight writes and validates staging content before a non-overwriting rename into the final location.

Option	Default	Accepted values	Behavior
id	required	1-64 letters, digits, _ or -	Stable destination identifier
folder	required	Absolute local or UNC path	Existing final intake directory
stagingFolder	required	Absolute local or UNC path	Separate non-overlapping directory on the same filesystem root
naming	PreserveStem	PreserveStem, JobId	Final basename policy

With `PreserveStem`, folder-acquired documents use their original basename with `.pdf` or `.tif`. Scanner capture has no original source pathname, so its filename uses the job ID. `JobId` always uses the job ID. A conflicting final filename is held; Paperweight does not overwrite it or invent a suffix.

Set up delivery and import

Make the final destination exist before use. Staging can be created by Paperweight if the operator has permission. Ensure the receiving application ignores staging and only ingests complete final files. Local drive letters must match; for UNC destinations use sibling folders on the same share. Passing the root check does not establish actual SMB atomic-rename behavior.

The signed-in operator needs permission to write and rename files in these folders. Paperweight records delivery to the folder; the receiving application handles import and filing.

Healthcare workflow targets

Paperweight delivers finished files to a local or network folder for the receiving workflow to import. The receiving workflow controls where those documents are filed and how staff access them.

Target	What to check in your workflow
TouchWorks EHR	Receiving import workflow, potentially through an intermediary document filer
Veradigm Practice Management	The site's supported document import workflow and naming/format requirements
Keena InteleFiler	SMB file intake, staging exclusion, filename contract, and downstream ingestion

Target	What to check in your workflow
DocuWare	File import, including the expected folder, document format, and metadata
OneDrive / SharePoint	Your upload or sync process sends finished files to the intended library or folder
Network shares and other targets	Accepted format/codecs, permissions, completion behavior, and naming contract

⊗ IMPORTANT

Keep Paperweight's working files and recovery copies in a protected folder on the PC, outside OneDrive or other synced folders. Use your organization's upload or sync process to send finished documents to OneDrive or SharePoint. Keep passwords and other credentials out of configuration files.

Limits and retention

Resource limits apply to isolated scanner/processor operations. Defaults are used when the policy's `limits` object is absent or null. An operation exceeding its budget is stopped or held; a higher value does not prove the endpoint can handle the workload.

Resource limits

Option	Default	Accepted values	Behavior
<code>maxInputBytes</code>	268435456	Positive integer bytes	256 MiB total acquired input/master budget
<code>maxPages</code>	500	Positive integer	Maximum pages per batch/document
<code>maxPagePixels</code>	100000000	Positive integer pixels	Per-page decoded pixel ceiling
<code>maxWorkerBytes</code>	2147483648	Integer ≥ 134217728 bytes	2 GiB default worker memory limit; minimum 128 MiB
<code>timeoutSeconds</code>	600	Positive integer seconds	Worker lifetime budget, default ten minutes

Paperweight currently holds a batch in memory inside the isolated worker. Test a batch as large as your team expects to handle, checking available RAM, disk space, and processing time. Lower resolution can reduce resource use but is a lossy policy decision requiring document review.

Retention is fixed in this version

`archiveDays` must be **7** and `historyDays` must be **90**. These JSON fields describe fixed supported policies; other values are rejected. They are not user-selectable retention durations in this release.

Successful jobs become eligible for archive expiration seven days after publication once cleanup has completed and their recorded state is Archived. Cleanup runs while the application is running; seven days is an eligibility boundary, not an exact deletion-time service guarantee.

⚠ WARNING

Archive expiration permanently deletes eligible local originals and job artifacts. Paperweight cannot recover these deleted files. Confirm that this retention period meets your needs before enabling live workflows.

Failed, incomplete, uncertain, and unexpected-content archives remain held for review. Linked directories, unexpected subdirectories, and unexpected artifacts prevent ordinary expiration. Investigate the reason instead of manually bypassing protection.

Activity events older than ninety days are removed. Records for expired jobs are removed when they have not been updated for ninety days. Unresolved job records remain available for review beyond that period.

Storage planning

Local storage needs include original masters, output, temporary processing files, and retained archives. Final document compression savings do not represent peak working-space savings. Account for incomplete and held work, which does not expire automatically, and monitor available disk space during normal use.

Uninstall preserves policy, local job stores, and archives. It is not a retention or cleanup operation. Establish a separate authorized process for any eventual data disposal.

Deployment options

Distribute the signed MSI with `Deploy-Paperweight.ps1`, `Detect-Paperweight.ps1`, and site-specific policy. The MSI installs the self-contained application per machine. Scanner drivers are a separate vendor deployment. Use the wrapper for maintenance on active endpoints so it can request a safe stop.

Manual MSI and deployment wrapper

Run from an elevated 64-bit PowerShell session in the deployment kit directory. Replace the illustrative publisher thumbprint with the approved release certificate thumbprint.

```
.\Deploy-Paperweight.ps1 -Action Install `
  -PackagePath .\Clinimatix.Paperweight.0.1.0.x64.msi `
  -ConfigurationPath .\configuration.json `
  -PublisherThumbprint RELEASE_CERTIFICATE_THUMBPRINT
```

For a fresh install where direct MSI deployment is appropriate:

```
msiexec /i Clinimatix.Paperweight.0.1.0.x64.msi /qn /norestart AUTOSTART=1 /L*v install.log
```

`AUTOSTART=1` is the default; `AUTOSTART=0` disables that MSI startup component. This setting controls startup at logon; Paperweight runs in the user session. Users operate scanners and shares in their own interactive sessions.

NinjaOne

Use [NinjaOne application automations](#). Upload the signed MSI, wrapper, detection script, and the appropriate configuration as the automation's helper files. Run the 64-bit PowerShell host as SYSTEM. Use the wrapper for upgrades/repair/uninstall; a fresh installation can also use the platform's native MSI installation action with quiet and no-restart arguments.

Run `Detect-Paperweight.ps1 -MinimumVersion 0.1.0` after deployment. Keep policy assignment scoped to the intended workstation/user. No NinjaOne credentials, API connection, or runtime service is required by Paperweight.

Microsoft Intune

Create an [Intune Win32 app](#). Package the deployment kit with Microsoft's Win32 Content Prep Tool. Configure the app with **System** install behavior, **x64** architecture, and **Windows 11** requirements. Use these commands:

```
powershell.exe -NoProfile -ExecutionPolicy Bypass -File .\Deploy-Paperweight.ps1 -Action Install -  
PackagePath .\Clinimatix.Paperweight.0.1.0.x64.msi -ConfigurationPath .\configuration.json -  
PublisherThumbprint RELEASE_CERTIFICATE_THUMBPRINT
```

```
powershell.exe -NoProfile -ExecutionPolicy Bypass -File .\Deploy-Paperweight.ps1 -  
Action Uninstall -PackagePath .\Clinimatix.Paperweight.0.1.0.x64.msi -  
PublisherThumbprint RELEASE_CERTIFICATE_THUMBPRINT
```

Use `Detect-Paperweight.ps1` as the custom detection script in the 64-bit host. It prints a detection string and exits zero only when the minimum registered version and executable are present. Update its minimum-version value in the deployment kit when creating a new application revision. Ensure the signed MSI remains available for wrapper maintenance commands, including uninstall.

Assign the app to the intended workstation groups. Use detection and return codes to track installation results.

Configuration Manager

Create a Windows Installer application or a script deployment type using the wrapper. Run as SYSTEM with version-aware detection. Separate the machine installation policy from the operator's access to scanners and destinations. Use a maintenance window for upgrades and verify return codes before declaring compliance.

Group Policy

Assign the MSI to computers at startup and deploy protected configuration separately. Avoid forced upgrades or reboots during capture; for active-session maintenance, first use the wrapper in an administrative maintenance task to drain work.

Return codes

Code	Meaning and response
0	Success
3010	Success, reboot required; schedule rather than force the reboot
1618	Busy/concurrent deployment or drain timeout; investigate and retry later
1603	Deployment failed; inspect the local installer log and policy validation

Other native Windows Installer errors are preserved. Configuration validation failure after a successful binary installation does not automatically roll back the installed binaries. Check both installed version and active policy.

Rollout and maintenance

Deploy consistent output profiles and source/destination assignments using your existing endpoint administration tools. Manage policy updates independently of application binaries. Some scanner vendor settings must be configured at the workstation.

Plan your rollout

Start with a workstation or small group, then expand by team or location. Assign someone to manage configuration and someone for operators to contact when they need help. Include scanner drivers, folder permissions, storage capacity, and retention in your deployment plan.

Before wider deployment, scan a few representative documents and follow them into the receiving application. Check readability, page completeness, and filing location with the staff who use them. Keep the working configuration so you can apply it consistently across the fleet.

Update configuration separately

Use `-Action Configure -ConfigurationPath ...` from the deployment wrapper. Retain the previous administrator-managed policy and record what changed. New policy affects newly acquired jobs; existing jobs retain snapshots. The wrapper's Configure action does not drain the application like an MSI maintenance action.

To restore policy, validate a copy of the known-good previous configuration and deploy it through the same Configure action. Do not edit user manifests or databases to make old jobs appear to use a new profile.

Upgrade, repair, and uninstall

The wrapper creates a protected maintenance marker and waits up to twelve minutes for Paperweight desktop/worker processes to exit. The application finishes the active processing cycle before servicing maintenance and cancels capture into retained incomplete state if needed. It does not force-kill busy processes. Concurrent wrapper operations return 1618.

Use `-Action Upgrade, Repair, or Uninstall` with the approved MSI and publisher thumbprint. The wrapper normally removes its maintenance marker when it finishes. If an administrative session was terminated unexpectedly, investigate any remaining marker; it keeps processing stopped until the maintenance issue is resolved.

There is no self-updater. Plan how the user will reopen Paperweight after maintenance or at the next logon, then confirm source health and retained work.

Rollback and preserved data

Uninstall preserves ProgramData policy and per-user operational data. Before returning to an older application version, confirm that it can read the data from your installed version. Older software rejects data formats it does not support. The MSI blocks installation of a lower product version through its normal upgrade path.

Never restore an old queue snapshot into live processing without reconciling downstream delivery. It can redeliver documents that were already consumed. Investigate using an isolated protected copy, and test your recovery steps there before applying them to live work.

Support your users

Let staff know which workflows to select, where finished documents go, and how to get help with held or incomplete batches. Use [Activity](#) to follow processing results and [Operations and recovery](#) when work needs attention.

Operations and recovery

Assign responsibility for held jobs, source health, storage capacity, and downstream reconciliation before enabling unattended intake. The app operates in the signed-in user's session and has no central fleet monitoring service.

Daily checks

Review Activity for Held, Uncertain, or persistently Published jobs and inspect source health. Confirm documents appear in the receiving application's actual workflow. Watch working-space growth, especially incomplete/held archives that do not expire automatically.

Keep Paperweight's local data on a protected fixed disk on the workstation. Network shares and synchronized folders are not suitable for the local job store.

Restart recovery

Interrupted state	Recovery behavior
Processing	Returns to Queued for another processing attempt
Acquiring	Held for incomplete acquisition/capture review
Publishing	Compare recorded final path/hash; ambiguous or missing output becomes Uncertain
Validated	Retained result can proceed to publication
Published	Source cleanup/archive can continue

An expected final file with the recorded hash can confirm publication after interruption. Its absence cannot prove that publication failed: the consumer may have already ingested and removed it.

Held documents

Select a held job in Activity and investigate its error. **Retry held document** applies only to eligible Held jobs. It is blocked if a publication intent exists, preventing blind redelivery. Incomplete scanner capture uses **Resume incomplete capture** instead.

An interrupted folder acquisition can retry only while the source still has its recorded identity and hash. If it changed, arrange a deliberate new submission rather than editing recovery metadata.

Uncertain publication

WARNING

Do not blindly retry, copy the document into intake again, delete publication metadata, or restore an old queue. Reconcile with the downstream operator first; duplicate clinical documents can result.

Compare the protected recorded publication intent and the receiving system's intake evidence under an authorized support process. The application has no automatic EHR-accepted status and no general-purpose UI command that safely resolves every uncertain publication.

Protected material and support

Original content, job snapshots, source paths, scanner preferences, and processing artifacts remain local. ACLs allow the current user, SYSTEM, and Administrators. This does not itself establish disk encryption, compliant backups, or safe crash dumps.

When requesting help, use sample documents without patient information whenever possible. Share only the approved diagnostic details needed for support, such as your product/configuration revision and error code. Keep patient documents, source paths, account secrets, and raw job data out of public tools. The installer log is `%ProgramData%\CLinimatix\Paperweight\installer.log`; handle it according to your organization's information-sharing policy.

See [Limits and retention](#) for archive deletion schedules and [Rollout](#) for maintenance and rollback guidance.

Troubleshooting

Find your error code or symptom below to choose the next step. Check the document and configuration before changing processing settings; allowing more transformations can change the resulting document.

Symptom or code	Check and next action
Inactive / no workflow	Valid managed policy, enabled source, ownerMachine and ownerUser matching the interactive session
ConfigurationRejected	Unknown properties, required arrays/IDs, invalid profile settings, overlapping roots; validate a corrected policy before deployment
UnsupportedPolicyVersion	Configuration schema must be 1, archiveDays 7, historyDays 90, revision ≥ 1
SourceOwnerRequired	Enabled source needs both workstation and Windows account ownership
OverlappingRoots	Separate source, destination, staging, and data-root directory trees
ProtectedStorageConfirmationRequired	Clinical mode is enabled without storage confirmation. Ask your administrator to review storage protection, then record the result with <code>protectedStorageConfirmed: true</code> . See Clinical mode .
ProfileNotQualified	At least one profile has <code>validatedForClinicalUse: false</code> while clinical mode is enabled. Ask your administrator to confirm the profile's document quality and record the result. See Clinical mode .
No scanners / ScannerUnavailable	Vendor driver installation, device connection, current-user access, and selected provider
ScannerCapabilityNotReported	The driver has not reported the required settings. Use the scanner's settings dialog and test a sample scan.
FeederEmpty / PaperJam / ScannerDisconnected	Review retained pages, recover the physical condition, and confirm completeness before finishing
LocalStorageRequired / SynchronizedStorageRejected	Move planned spool to protected fixed local storage using an authorized recovery procedure
AccessDenied / FileIoFailure	Actual operator permissions, locks, disk space, path availability, and endpoint controls
DestinationUnavailable	Final folder exists and is reachable by the operator
DestinationCollision / StagingCollision	Investigate existing file identity and delivery history; do not overwrite blindly
PublicationUncertain / ReconcileWithInteleFiler	Confirm delivery with the receiving system before resubmitting; either code can appear with any destination
PublicationReviewRequired	A publication intent blocks retry to avoid duplicate delivery

Symptom or code	Check and next action
ResumeCaptureRequired	Resume an incomplete scanner batch rather than retrying it as a completed document
SourceChangedResubmit	Original folder file identity/content changed; arrange a deliberate new submission

Content or profile holds

Encrypted and active PDFs are held. Signature-bearing PDFs require eligible unchanged passthrough. Metadata or complex-PDF transformations can be held when the requested policy cannot be safely established. Codec restrictions do not make an unsupported encoder available.

Page, input-byte, pixel, output-size, or resolution errors require comparison with the approved profile and the actual document. Resource errors may indicate an oversized batch or endpoint capacity problem. If you split a batch or scan it again, follow your team's recovery steps and check that every page is included in the correct order.

For an unsupported file, keep the original intact. If possible, provide support with a small sample that contains no patient information and explain the result you expected.

Maintenance remains active

If Paperweight exits or stays stopped after a failed administrative session, check for the protected `maintenance.flag` file and the installer log. Establish that no installation is active before correcting a stale marker through an administrator. Do not kill processing workers to make an upgrade proceed.

Support context

Provide product version, documentation revision, policy/profile revision, acquisition type, error code, and whether delivery is confirmed or uncertain. Describe the scanner/provider or target contract without transmitting patient information. Clinimatix contact: hello@clinimatix.com.

Commands and scripts

Use these commands to start Paperweight, validate settings, or manage an installation. Quote paths containing spaces. Keep passwords and tokens out of configuration files.

Desktop executable

Invocation	Behavior
No arguments	Open the window with managed policy and default local data root
<code>--tray</code>	Start with the window hidden in the notification area
<code>--validate-config <path></code>	Validate configuration without starting the interactive app; exit 0 or 2
<code>--config <absolute-json-path> --data-root <absolute-local-directory></code>	Use separate configuration and local storage for an isolated evaluation, in this argument order

The custom configuration/data-root command cannot be combined with `--tray`. Unsupported combinations return exit code 2. Only one interactive instance per Windows user/session is allowed; starting it again does not open another instance.

Deploy-Paperweight.ps1

This script requires elevation. Use the deployment kit's version with its matching MSI.

Parameter	Default	Purpose
Action	Required	Install, Upgrade, Repair, Uninstall, Configure, or Detect
PackagePath	Unset	MSI path required for Install, Upgrade, Repair, and Uninstall
ConfigurationPath	Unset	Policy to deploy on Configure or after successful Install/Upgrade/Repair
ExpectedVersion	0.1.0	Minimum version used by wrapper Detect
PublisherThumbprint	Unset	Expected signing-certificate thumbprint for MSI operations
AllowUnsignedDevelopment	false	Explicitly bypass publisher verification for isolated development testing

Always supply `ConfigurationPath` for `Configure`; omitting it currently performs no configuration update. Wrapper `Detect` checks the registered version. Prefer the standalone detection script for endpoint management because it also checks that the installed executable exists.

Examples beyond [installation](#):

```
.\Deploy-Paperweight.ps1 -Action Upgrade `
  -PackagePath .\Clinimatix.Paperweight.0.1.0.x64.msi `
  -PublisherThumbprint RELEASE_CERTIFICATE_THUMBPRINT

.\Deploy-Paperweight.ps1 -Action Repair `
  -PackagePath .\Clinimatix.Paperweight.0.1.0.x64.msi `
  -PublisherThumbprint RELEASE_CERTIFICATE_THUMBPRINT

.\Detect-Paperweight.ps1 -MinimumVersion 0.1.0
```

Detect-Paperweight.ps1

`MinimumVersion` is a .NET version value, default `0.1.0` in this kit. Exit 0 plus a printed detection string means the registered version is sufficient and the executable exists; exit 1 means not detected. Run in the 64-bit host to inspect the expected registry view.

Direct MSI maintenance

```
msiexec /fa Clinimatix.Paperweight.0.1.0.x64.msi /qn /norestart /L*v repair.log
msiexec /x Clinimatix.Paperweight.0.1.0.x64.msi /qn /norestart /L*v uninstall.log
```

These direct commands do not provide the wrapper's drain coordination. Use the wrapper on active endpoints. See [Deployment options](#) for return codes and [Rollout](#) for recovery.

Release notes and limits

0.1.0 public preview

Documentation revision **2**, published **2026-09-12**.

With this preview, you can capture and review scans, process documents from watched folders, and create consistent PDF or TIFF output using centrally managed settings. Paperweight keeps originals locally for recovery, records processing activity and file sizes, and helps prevent accidental overwrites or duplicate delivery after interruptions. Administrators can deploy it with an MSI and their existing endpoint tools, including NinjaOne, Intune, Configuration Manager, and Group Policy.

Preview status

Paperweight is still in development. Features and supported configurations may change as we incorporate feedback. Share problems or suggestions with your Clinimatix contact, including what you were doing and the result you expected.

Current limitations

This version does not include OCR, clinical classification, direct EHR APIs, native cloud connectors, a Windows service, central reporting, or automatic updates. Cross-team activity reporting and trends are planned. To use Paperweight with a healthcare application or cloud destination, use the file-based intake workflow described under [Destinations](#).

Automatic page rotation is not supported. Paperweight may hold complex or signed PDFs and documents with unsupported metadata or color profiles for review. Successful local archives are retained for seven days and activity history for ninety days; these periods are fixed in this version. A Published status confirms file delivery, not acceptance by an EHR.

Glossary

Term	Meaning
Acquisition	Creating protected originals from a scanner or completed folder input
Archive	Local recovery material belonging to a successfully published and cleaned-up job
Completion contract	Producer behavior that establishes when a folder file can be acquired
Destination	Final folder plus its separate staging folder and filename policy
Held	Work retained because a failure or review condition prevents progress
Immutable job	Acquired content and policy snapshot retained as the basis of processing/recovery
Normalization	Applying a permitted document output policy and validating the result
Owner	Windows workstation and user assigned to execute a source
Profile	Versioned output format, transformation permissions, and validation restrictions
Published	Filesystem handoff completed; distinct from downstream ingestion
Ready marker	SHA-256-bearing sidecar that identifies the completed source bytes
Source	Scanner or folder workflow linked to a profile and destination
Spool / job store	Protected per-user local working state, originals, and recovery files
Staging	Temporary destination-side area excluded from consumer ingestion
Uncertain	Publication may have happened; reconciliation is required before further action
Workflow target	Application or process that receives your finished documents, such as an EHR or document management system

Third-party notices

Copyright © Clinimatix, LLC. Paperweight application code is proprietary. Third-party components retain their own licenses. This documentation website does not distribute Paperweight application binaries.

Application components

Component	Baseline	Role / license context
.NET Windows Desktop runtime	10.0.12	Self-contained runtime; Microsoft distribution and third-party notices
NAPS2 SDK, WPF images, Win32 worker	1.3.0	Scanner/image/PDF integration; LGPL-2.1-or-later
NAPS2 PDFium binaries	1.3.0	PDF rendering; PDFium and bundled native notices
Microsoft.Data.Sqlite	10.0.12	MIT with SQLite/SQLitePCLRaw transitive notices
System.Diagnostics.EventLog	10.0.12	Optional local Windows audit lookup; MIT
qpdf	12.4.1	Apache-2.0 and bundled dependency notices

Open the installed application's **About** tab to find third-party notices, font licenses, and the complete bundled legal materials, including the applicable LGPL source and replacement information.

Website and manual

Manrope and JetBrains Mono are bundled locally under the SIL Open Font License 1.1. [Manrope license](#) and [JetBrains Mono license](#) are distributed with this site. Canonical Clinimatix brand assets retain their owner's rights.

The site and manual use DocFX and its modern template under the [MIT license](#).

Upstream references

- [NAPS2 source](#)
- [PDFium license](#)
- [qpdf source and notices](#)
- [DocFX](#)

Product and company names identify intended receiving workflows; their appearance does not imply endorsement, certified integration, or a native connector.