

StreamGate Windows User Guide

StreamGate 1.9.7 · Customer delivery document

StreamGate is a camera ingest and distribution gateway for private deployments. The Windows portable package is suitable for evaluation, laboratory verification, and smaller deployments that do not require an installation wizard. The Windows installer provides the same runtime with a managed Windows service lifecycle.

This guide applies to StreamGate 1.9.7. A backup always includes the data directory and may also include selected tasks' sealed recording directories. Stop Gateway, Core, and other recording writers before creating a backup; copying and verifying more recordings extends downtime. After restoring, check recording paths, indexes, and retention before releasing task protection and enabling recording. The recording timeline links only saved events to complete segments with verified UTC sidecars.

Earlier upgrades: the one-time 1.9.6 settings migration

Stop the old version and retain its data directory. Use the installer to upgrade an installed deployment. For portable packages, replace program files while preserving [data](#), licenses, and recordings.

On the first upgrade from a version earlier than 1.9.6, StreamGate backs up the configuration database and restores system media policy and each channel's on-demand ingest, playback output, and transcoding settings to the 1.9.6 defaults. On-demand ingest is off by default. Devices, channels, and main/substream input sources are retained, as are network and CORS settings, accounts, API tokens, recording configuration, and license identity. Back up the existing data directory before upgrading.

This adjustment runs once. Later 1.9.6 restarts and future releases do not repeat it. The original configuration backup is stored under [data/backups](#). After startup, select any required playback protocols, transcoding, and substream output policies in System Settings and the channel editor.

Package contents

- [StreamGate.exe](#): public application entry point and service-control command.
- [streamgate-helper.exe](#): independent executor for backup and restore maintenance jobs.
- [core/streamgate-core.exe](#): local media runtime.
- [sidecars/mediamtx/](#): bundled media distribution service.
- [configs/](#): default configuration files.
- [runtime/ca/](#): public CA trust bundle used to validate HTTPS, RTMPS, and RTSPS inputs.
- [data/](#): local database and generated deployment identity in a portable deployment.
- [logs/](#): runtime logs in a portable deployment.
- [licenses/](#): license and public-key files.
- [使用说明.pdf](#): Simplified Chinese guide for the current platform.

- [docs/zh-TW/使用說明.pdf](#): Traditional Chinese guide for the current platform.
- [docs/en-US/StreamGate-User-Guide.pdf](#): this English guide.
- Professional and Enterprise packages additionally contain the three localized Public API PDFs and [docs/streamgate-public-api-v1.openapi.yaml](#); Standard packages contain no API document.
- [install-service.bat](#), [uninstall-service.bat](#), [start-service.bat](#), [stop-service.bat](#), and [service-status.bat](#): optional Windows service helpers in the portable package.

The Web Console is embedded in [StreamGate.exe](#); no separate web directory is required. The HTML API reference and Postman collection are optional integration aids and are not included in runtime packages.

Analysis Node is an optional component distributed and installed separately. It is not embedded in the StreamGate installer or portable package. A deployment that only uses ingest, distribution, recording, playback, and the Public API does not need Analysis Node. For intelligent analysis, install StreamGate first, then install Analysis Node 1.3.x for the same platform and select either local attachment or remote-node mode. CUDA/OpenVINO accelerator packages and [.sgcap](#) capability packages are also delivered separately.

Quick start

For a desktop evaluation, double-click [StreamGate.exe](#). The application starts the local backend and opens the Web Console.

Right-click the StreamGate icon in the Windows notification area for desktop lifecycle controls. Exit StreamGate (stop local backend) stops the backend launched by this portable package before closing the tray. Exit tray only closes the icon and leaves the backend running. Older releases with separate actions must use Stop local StreamGate followed by Exit tray.

For unattended operation from the portable package, run [install-service.bat](#) as an administrator. The service starts immediately and is configured for automatic startup with Windows.

Web Console addresses:

On the StreamGate server: <http://127.0.0.1:18080/>
 From another computer: <http://<server-lan-ip>:18080/>

On first use, complete the account setup in the Web Console. Do not expose the management port directly to an untrusted network.

Portable-package service scripts

Run these [.bat](#) files from the extracted portable-package directory. The Windows service records the current absolute package path, so move the package to its final directory before installing the service.

Script	Administrator required	Purpose
install-service.bat	Yes	Installs and starts the StreamGate Windows service from the current directory. Requests UAC elevation when required.

Script	Administrator required	Purpose
service-status.bat	No	Reports whether the service is installed and running and shows the default Web Console address.
start-service.bat	Yes	Starts the installed StreamGate service.
stop-service.bat	Yes	Stops the service without deleting data or configuration.
uninstall-service.bat	Yes	Stops and removes the service while preserving data/ , logs/ , licenses/ , and configs/ .

Typical portable service workflow:

1. Extract the package to its final application directory.
2. Right-click [install-service.bat](#) and choose Run as administrator.
3. Open <http://127.0.0.1:18080/> on the server and complete first-use setup.
4. Run [service-status.bat](#) to confirm the service state.
5. Use [stop-service.bat](#) and [start-service.bat](#) for maintenance.
6. Run [uninstall-service.bat](#) before moving the package or retiring the service.

Installer behavior

The Windows installer registers one managed [StreamGate](#) service. Reinstall and repair preserve the existing data directory, configuration, deployment identity, license files, recordings, and separately managed Analysis Node installation.

If a StreamGate service already exists, the installer validates its ownership and installation record before replacing it. It will not silently take over an unrelated or inconsistent service. Use the current StreamGate installer for repair instead of manually deleting service or registry state.

The installed public entry point remains [StreamGate.exe](#). Double-clicking it starts or connects to the installed service and opens the Web Console; it does not create a second portable instance with a separate database.

Camera ingest and playback settings

After adding an ONVIF device or RTSP channel, check the input codec and resolution. The channel main stream uses the selected input, which may be a camera substream; it is not necessarily the camera's physical main stream. Stream & Output has separate Main stream and optional Substream settings. Without a substream, only the main stream is available.

In System Settings > Media & Playback, Console playback stream sets [media.defaultViewingQuality](#): [auto/high/fluent](#) mean Auto, Main stream, and Substream. This applies to console preview, details, and Standard/Professional video walls. Auto selects by window size and presentation; an unavailable substream falls back to main. The channel UI no longer has a default-playback selector. Legacy [viewing.defaultQuality](#) remains only for Public API playback-URLs requests without [quality](#); explicit [quality=high/fluent/auto](#) still overrides it. The system console choice does not change Public API behavior, recording sources, or GB28181 delivery.

The main stream offers Use system, Source stream, Compatibility, and Always transcode. Substream sources are Device substream Profile, Device substream URL, and Transcode from main input. A device substream also has its own four output modes and parameters. Changing one stream does not overwrite the other. Save changes to apply them; they do not alter the camera's encoding.

Transcode from main input offers only Compatibility and Always transcode. A new configuration defaults to Always transcode, preserving source dimensions/frame rate and inheriting the system preset; explicitly set substream parameters when lower bandwidth is needed. It shares the original input with the main stream and encodes independently; it does not re-encode the main stream's encoded output. It can reduce gateway-to-viewer traffic, not camera-to-gateway ingest traffic.

- Compatibility converts only H.265 to H.264 and passes H.264 through. Always transcode continually encodes. Both allow independent bitrate, dimensions, frame-rate and GOP settings; the encoder backend is global. Compatibility ignores these parameters when conversion is unnecessary; it does not transcode H.264 merely to resize it. Transcoding uses CPU/GPU resources; check Actual Output for the active codec and backend.
- Transcoding Keyframe interval (ms) is specified in milliseconds; an empty field or 0 inherits the global interval (default one second). Shorter intervals can increase bitrate or storage. Passthrough does not change camera GOP; configure source GOP on the camera or NVR.
- On-demand ingest enabled means the source is pulled only while viewing, recording, analysis, or another task needs it, and can stop after becoming idle. Cold first frames may take longer. On-demand ingest disabled keeps the source connected while the channel is enabled. Active recording or analysis can keep the source connected even with on-demand ingest enabled.
- LL-HLS pre-generation under System Settings > Media & Playback offers Off, Selected channels, and All always-on channels. It is off by default and applies only to enabled, always-ingested channels. Continuous packaging uses CPU and memory caches, not disk by default; on-demand channels remain on demand.

For faster first frames, disable on-demand ingest, use browser-decodable H.264/AAC, and evaluate a roughly 1-2 second source keyframe interval on the camera. To reduce camera uplink traffic, enable on-demand ingest, lower device bitrate, or use a device substream. Viewing the main stream and device substream together may open two camera inputs. If only one lower-bitrate stream is needed, use the device substream as the channel input.

In System Settings > Media & Playback, choose Save bandwidth, Balanced or Prefer quality. The adjacent gear opens Transcode settings for each preset's quality strength and optional bitrate limit, plus the shared keyframe interval. Select the global encoder backend under Advanced playback & resources below.

Automatic presets preserve source resolution and frame rate. Set dimensions or a frame-rate limit explicitly in the channel's main/substream output when needed. Main and substream Use system independently inherit the global transcoding mode and preset; enabling transcoding does not create an unconfigured substream.

Preset	Default quality strength	x264 CRF	Intel QSV QVBR quality
Save bandwidth economy	4	28	28
Balanced balanced	6	25	25

Preset	Default quality strength	x264 CRF	Intel QSV QVBR quality
Prefer quality quality	8	22	22

Higher quality strength preserves more detail and usually needs more bitrate. Strength is not a native encoder value: x264 uses CRF+VBV, QSV uses QVBR when supported, and NVENC/AMF use their supported quality-priority VBR modes. OpenH264 and other backends without the corresponding mode fall back to a bitrate budget, not equivalent CRF quality. Backend values do not imply equal quality or recovery of detail already lost at the source.

The three [economy/balanced/quality](#) presets are stored in [media.videoTranscodeParameters.automaticPolicy](#). Each [qualityLevel](#) is an integer 1..10; zero/omission uses the built-in value, with nondecreasing preset order. Optional [maxBitrateKbps](#) accepts zero or an integer 256..64000 kbps. Zero/empty lets the system estimate a budget. A limit is neither a fixed bitrate nor a hard network cap including audio, packaging and retransmission. Unsupported quality modes are reported as [bitrate-fallback](#) in actual output.

An omitted, empty or all-zero channel [video.automaticPolicy](#) inherits the whole system policy; a nonzero object replaces it, using built-in defaults for missing fields. A positive channel [bitrateKbps](#) selects manual bitrate and bypasses automatic presets. All transcoding outputs use the global encoder backend; legacy channel [encoderPreference](#) remains readable but does not override it. A zero channel keyframe interval inherits the global value, whose default is 1000 ms.

Compatible mode passes an H.264 device stream through and transcodes H.265 to H.264. Transcode from main input explicitly creates an independently encoded output even when the main source is H.264. These source types differ; the generated substream does not re-encode the main output's encoded data.

Legacy [sourcePercent/referenceBitrateKbps/referenceCeilingKbps](#), [automaticBitrate](#) and [hevcBitrateReservePercent](#) remain accepted for internal budgeting/fallback, not quality strength. Budgets consider measured source bitrate and actual output dimensions/frame rate; before measurement they use a format estimate. Legacy [maxWidth/maxHeight/maxFps](#) inside automatic presets no longer change the picture. Explicit channel [video](#) geometry and frame rate remain effective.

Cancel leaves the page unchanged; Apply edits the draft, and Save system settings makes it effective. Applying without edits creates no unsaved state. Shared-encoding edits preserve a legacy fixed bitrate; editing the automatic policy or choosing Use automatic bitrate switches to automatic mode. Read runtime state for the adopted backend, rate-control mode and native quality value. No Public API system-settings endpoint is added.

Saving a global preset or policy may briefly reconnect active video outputs that use the changed value. Independent channel parameters, explicit policies, and manual bitrates are not overwritten; unused preset changes do not rebuild the current output. Codec or dimension changes recalculate targets. Significant source frame-rate changes also update the automatic estimate after independent stable windows and a cooldown, without requiring a large source-bitrate change; small frame-rate jitter does not trigger adjustment. Short VBR fluctuations do not trigger second-by-second changes; sustained large changes require stable windows and a cooldown before adoption at a keyframe boundary. Reconnect a device that does not report its new codec. Public API players must still handle stream endings, track changes, and stalled playback.

In older configurations without a separate substream output policy, device Profile/URL substreams share the main stream's output policy. Generated substreams use the old output policy while the main stream uses the system policy. Viewing does not rewrite the configuration. The first editor save creates independent policies while preserving the previous behavior of both streams. An explicit Use system selection for the device substream makes it inherit the system independently.

In the Channel tab, set Recording stream to Main or Substream, then choose Original data or Transcoded data under Recording data. Main means the channel's main stream, not necessarily the camera's physical main stream. Substream is unavailable when not configured. An independent device substream can record its original data or its output; a generated substream has no separate original input and displays and uses only its output. Transcoded data follows the selected stream's actual output. When transcoding is unnecessary, that output is saved directly and the choice is retained.

New tasks default to the main stream's original data. Older tasks without a recording-stream selection use Main. Opening a task does not rewrite its settings; System processing (existing setting) retains its previous behavior until a new recording target is explicitly chosen. If changing the substream source or disabling it invalidates a target, choose a valid target or turn recording off before saving.

The Channel and Recordings pages manage the same task; selecting a recording target does not change its switch or schedule. Turning recording off preserves the schedule; turning it back on resumes it. Manage schedules, segments and retention in Recordings. Deleting a device retains its channels. Deleting a channel removes recording tasks and indexes, not disk files; they disappear from the playback list. To pause recording, use the switch.

GB28181 upstream cascading

Set Stream mode (all cascaded channels) under GB28181 > Upstream cascade. Single stream uses Main by default, or Substream for all channels. Dual stream provides both channel streams to the upstream platform. The policy applies to all locally published channels; there are no per-channel overrides. A channel without a substream uses Main without blocking cascading. Main means the channel's main stream. Substream can use a device substream or be transcoded from the main input. Each stream uses its own saved output parameters.

Saved policy is displayed separately from draft selections. Active sessions determine the actual stream in use. Older configurations without a stream mode display Keep existing stream handling. Opening the page or saving other fields retains that behavior. A new policy is adopted only after explicitly choosing and saving a stream mode; the upstream platform then needs to request playback again.

Upgrade

Before upgrading a desktop portable deployment, use Exit StreamGate (stop local backend) in the old tray. For releases such as 1.8.2, use Stop local StreamGate and then Exit tray. If an older release still owns the management port, the new launcher reports the version that is actually running instead of opening the old console as though it were the new release.

For a portable service deployment, stop the service, back up the current directory, and replace [StreamGate.exe](#), [core/](#), [runtime/](#), [sidecars/](#), [docs/](#), and the root [使用说明.pdf](#). Do not overwrite [data/](#),

`logs/`, `licenses/`, or `configs/`. Do not overwrite an `analysis-node/` directory managed by the separate Analysis Node installer.

For an installed deployment, run the new installer. Keep the existing data and license locations unless an approved migration requires different paths.

TLS media inputs always validate the certificate chain and host name. If an enterprise deployment uses a private CA, set `STREAMGATE_SOURCE_TLS_CA_FILE` to the enterprise PEM trust bundle before starting StreamGate.

Ports and firewall

The default Web Console and API listener is `0.0.0.0:18080`. LAN access requires TCP port `18080` to be allowed on the StreamGate server. Client computers do not need a matching local port; they connect to `http://<server-lan-ip>:18080/`.

Media protocols may require additional ports according to the enabled StreamGate configuration. Open only the protocols that the deployment uses. When a reverse proxy terminates HTTPS, keep the direct management listener on a trusted network.

Logs and diagnostics

Use the Web Console log page for routine troubleshooting and diagnostic export. Runtime logs identify the StreamGate version, component startup, configured listeners, and recoverable errors without exposing full credentials. Rotate or remove old diagnostics according to the site's retention policy; do not delete the active database, deployment identity, license, or recordings as part of log cleanup.

For a startup failure, record the displayed error, the installation path, service state, Web Console address, and relevant log time range. Avoid publishing API tokens, camera credentials, license files, or complete private URLs in screenshots or support tickets.

Licensing

The public Standard and Professional runtime packages both provide a long-term free baseline of five input channels when no license is installed. Professional capability evaluation requires a signed trial license. Evaluation video has no watermark; term, capacity, and feature scope follow the issued license. Paid capacity, Professional features, and commercial rights require a matching license.

Standard capabilities include GB28181 registration and keepalive, catalog publishing, live media requests, and media sessions; advanced workflows require the applicable Professional or Enterprise entitlement. Public REST API, WebHook, and MQTT are unavailable in Free and Standard. A valid Professional trial, Professional, or Enterprise software license enables them, subject to the `openapi` entitlement and runtime package. REST access and REST integration management also require the corresponding Token scopes; event receivers use WebHook signatures or MQTT credentials separately. Analysis licensing does not replace these software entitlements. Professional trials are only for permitted non-production evaluation.

The license page shows the deployment identity and enabled features. A StreamGate license does not automatically grant Analysis Node or individual analysis-capability entitlements; those rights are evaluated independently.

Offline cold migration from Windows 1.8.2

The supported entry points in the new package are [StreamGate.exe migrate plan|prepare|verify](#). Help is available on Windows and Linux, but migration execution is supported only on Windows. The migration commands do not elevate privileges, stop processes or services, change the registry or firewall, run the old uninstaller, or clean a failed copy.

Do not resolve an old Inno AppId conflict by running the old uninstaller. Preserve the registered installation and use a separately reviewed non-destructive recovery procedure, or use a clean Windows host.

Before migration, identify the old executable, working directory, stopped state, and every data, recording, and Node root. Replace every placeholder in this layout with verified paths and the reviewed SHA-256 of the old executable. Remove unused roots. Program Files and ProgramData may remain separate, and roots may be on different drives. Relative paths are resolved against [workingDirectory](#); remote TLS paths are resolved against the configuration-file directory. Every root must be an explicit, non-overlapping local application directory without links. The process never scans an entire drive.

```
{
  "schema": "streamgate-cold-layout-v1",
  "kind": "gateway",
  "sourceRoot": "C:\\Program Files\\StreamGate",
  "workingDirectory": "C:\\ProgramData\\StreamGate",
  "dataDirectory": "C:\\ProgramData\\StreamGate\\data",
  "databaseFile": "C:\\ProgramData\\StreamGate\\data\\streamgate.db",
  "deploymentFile": "C:\\ProgramData\\StreamGate\\data\\deployment.id",
  "licenseFile": "C:\\ProgramData\\StreamGate\\licenses\\license.lic",
  "licensePublicKey": "C:\\Program Files\\StreamGate\\licenses\\public.pem",
  "machineld": "",
  "sourceExecutableSHA256": "<REVIEWED_OLD_EXE_SHA256>",
  "stopped": true,
  "additionalRoots": [
    {"sourceRoot": "C:\\ProgramData\\StreamGate", "targetRelativePath": "state"},
    {"sourceRoot": "E:\\StreamGateRecordings", "targetRelativePath": "recordings"}
  ]
}
```

[machineld](#) is verified context from the old license, not a generated substitute. Machine-bound licensing still requires a separate runtime acceptance check. Deployment identity, license and public key, account password hashes, and history are preserved. Layout, plan, and receipt field names are case-sensitive; unknown fields are rejected. Review the plan and use its emitted SHA-256. If candidate bytes change, create a new plan and use a new target.

```
& 'F:\StreamGate-candidate\StreamGate.exe' migrate plan -layout  
'F:\migration-inputs\layout.json' -target-root 'F:\StreamGate-new' -output  
'F:\migration-inputs\plan.json'  
& 'F:\StreamGate-candidate\StreamGate.exe' migrate prepare -plan 'F:\migration-inputs\plan.json'  
-plan-sha256 '<PLAN_SHA256>'  
& 'F:\StreamGate-candidate\StreamGate.exe' migrate verify -receipt  
'F:\StreamGate-new\migration\receipt.json' -receipt-sha256 '<RECEIPT_SHA256>'
```

`plan` writes only the explicitly named plan and does not create the target. `prepare` creates a new private copy and preserves original files under `.migration/source`. `verify` is read-only: it does not add files, update schemas, or start StreamGate. On failure, the original store, directories, evidence, and partial target remain in place. Do not retry into the same target or edit a pin manually.

The original SQLite files are read as bytes. Committed WAL content is materialized only in a separate working copy; the original DB, WAL, and SHM remain unchanged. An uncleared rollback journal is rejected. Preserve all sidecars, then use the verified old version to recover and stop normally under a separately authorized operation. Never delete the journal or edit the database header to bypass the check.

Only the declared primary SQLite database is migrated. The verifier proves legacy column types, defaults, keys and constraints, and SQLite value types. New schema elements are allowed; silently dropping legacy values is not. Active and rollback capability packages are signature-verified. Historical packages that do not meet current admission remain only in `.migration/source`, the signed archive, and the receipt; they do not enter runtime `packages` or `previousVersion` directories.

`offlineVerified=true` does not confirm activation. Keep the old instance and backup; start with mapped data, database, and license paths, then check service ownership and runtime behavior.