CtxLens

CtxLens Reports

Three distinctly different reporting paths in CtxLens.

1. The three paths

CtxLens serves three distinct relationships between a user and AI. Each is supported as a primary, complete use of the product. None is treated as the default. The Report Settings view, where every report begins, is built so the user's choice among the three is explicit, exclusive where exclusivity matters, and additive only where multiple layers would not produce incoherent presentation.

The three relationships, ordered by ascending AI involvement:

- A user who uses no AI at all — by preference, by company policy, or by contract.
- A user who uses on-device AI only — to gain summarization without exposing data to a third-party service.
- A user who uses external AI — to produce complex deliverables, or to populate reporting templates, with a third-party large language model.

This document describes each path in turn. The order is deliberate: a reader can read the section that applies to them and stop. The first path uses CtxLens with no AI involvement at all. The second adds on-device summarization to the first and is described as an extension of it. The third is structurally different from the first two and is described independently.

A note on workflow. Before the user reaches Report Settings, they have selected the card population — the subset of cards to include in a specific report. CtxLens provides several methods for this selection (filters, queries, and manual selection) and they are documented elsewhere. This document begins where the card population is chosen and the user is in Report Settings, where the path is selected and the report is generated.

A note on naming. CtxLens has a global *Settings* view (where the user configures app-wide preferences) and a per-export *Report Settings* view (where the controls described in this document live). This document refers to them by those names. A handful of report-time choices draw their behavior or defaults from the global Settings; those connections are flagged where they appear.

2. The promise

CtxLens makes a single, central commitment about user data, and the three paths described in this document each honor it without exception. CtxLens does not send data off the device . . . ever. There is no cloud sync. There is no corporate backend. There is no server that ever sees the user's work. There are no APIs in the app that transmit card content, photos, or any other user-generated material off the device — not for analytics, not for sync, not for backup, not for anything. The promise is structural. It is the absence of the capability itself, not a mutable policy.

A reader can be forgiven for wondering how the promise holds across paths that look quite different from one another. It is worth walking through each.

The first path — no AI. The user's cards remain on the device, the report is composed on the device, and nothing about the workflow touches a network. There is no model to consult, no service to call. The user receives a document and decides for themselves what to do with it.

The second path — on-device AI. Summarization is performed by Apple Intelligence, which runs locally on the user's device. Absolutely no content is transmitted to a server during summarization. The label beneath the AI Summary controls in Report Settings states this directly: *“Uses Apple Intelligence to generate a summary. Summarization is performed entirely on your device.”* CtxLens does not invoke any service that would override this. The summary is generated locally and inserted into the report locally.

The third path — external AI. This is the path that most needs to be examined, because it produces an output specifically intended for a third-party AI service. The promise still holds. The export is generated on the device. CtxLens does not transmit it. The user previews it on the device, and if they choose to share it, they share it themselves — through email, AirDrop, Save to Files, or whatever sharing channel they pick. From the moment of export onward, the user is the agent. CtxLens does not call any external model. There is no network code in CtxLens that uploads card content to anywhere. If the user takes the export to a third-party AI, that is the user's decision, and it is the user's action.

This is the difference between a tool that exposes data and a tool that produces an artifact the user can choose to expose. CtxLens is the second kind. The choice to share remains, in every case, with the user; when chosen, the sharing is completed without programmatic participation from CtxLens.

3. Scale

CtxLens reports can include hundreds of cards and thousands of images in a single export. This is unusual for an iOS app, and the reason it is unusual is worth stating directly: it shapes what the product can reasonably be asked to do.

iOS apps that build large documents in memory tend to fail at scale. The system kills any app whose memory usage exceeds a per-app ceiling that is much smaller than most users imagine — a few hundred megabytes — and a report that assembles a full DOCX or PDF in memory before writing it to disk hits that ceiling quickly when the report contains photos. This is why most documentation apps cap the number of images per report, recommend splitting reports across multiple runs, or simply crash when asked to do too much.

CtxLens does not assemble its reports in memory. It streams them. Every paragraph, table row, image reference, and zipped archive entry is written to disk the moment it is composed, which keeps the app's peak memory usage roughly constant regardless of report size. Each image is loaded, processed, and released in its own autorelease pool, so memory does not accumulate across the loop. The DOCX archive is built with a custom ZIP writer that supports the ZIP64 extension, so total report size can exceed four gigabytes without corruption. Background-task handling and a Live Activity keep the export running and visible even if the user switches away from the app while it works.

The user-visible consequence is straightforward. A monthly project log with three hundred cards and fifteen hundred images can be exported in one operation. A quarterly investigation with eight hundred cards and four thousand images can be exported in one operation. The user is not asked to split the work, prune the input, or restart the export halfway through. CtxLens shows an estimate before starting, asks for confirmation on large jobs, and then runs the export to completion. While a large export is running, the user can continue using every other function in CtxLens, and the rest of the device, without interruption. The only action prevented during this period is initiating another export, transfer, or backup.

Two practical thresholds matter to the user. A report whose embedded images push the document above 50 MB is delivered as PDF only — DOCX files at that size become unstable in Word and Pages, which is a property of the format itself rather than a CtxLens limitation. A report whose embedded images push the document above 1 GB is blocked outright, with a recommendation to deliver images in a separate folder rather than embedded; here too, the limit is a property of the format rather than the app, since PDFs above 1 GB can become unstable in some viewers.

Inside those thresholds, the product handles the load. Above them — when images are delivered as a separate folder rather than embedded — the only practical constraint on export size is the storage available on the user's device. Twenty-gigabyte packages and larger are well within CtxLens’s capability.

4. The first path: no AI

Some users have no interest in AI, by preference. Others are contractually prohibited from exposing their work, or their company's data, to AI of any kind. Others are bound by company policy to the same effect. CtxLens treats all of them the same: as primary users with full access to the product's reporting capability, requiring no engagement with AI at any point.

How the path is selected

In Report Settings, the user leaves both the *Report Optimized for use with External AI* toggle and the *AI Summary* toggle off. No data leaves the device. No model (local or remote) processes the cards. The report is composed by CtxLens directly from the selected card population and the user's formatting choices.

What the user controls

A note before the controls themselves. The defaults that ship with CtxLens are calibrated to suit most users most of the time. When a user changes any value in Report Settings, the new value becomes their default for that choice going forward. Choices persist across exports and are restored the next time the user opens the view. A user converges on their preferred report shape quickly, after which they typically open Report Settings, see the values they already prefer, and tap *Generate Report*. The detail below exists for users who want to understand what each control does. It is not a checklist to reconsider on every export, though changes can be made at any time.

Title and subtitle. Both are optional free-text fields. When provided, they appear on the first page of the report. When omitted, the report is generated without them.

Omit Entire Timestamp. When on, no date or time appears on any card. The cards still appear in chronological order, but the timestamps themselves are suppressed. The two controls below (Omit Times and Date-Grouped) are hidden when this is on, since they describe behavior that no longer applies.

Omit Times. When on, the date appears on each card, but the specific time of day does not. Useful for reports where the day matters but the time does not.

Date-Grouped. On by default. Cards are grouped under a date header rather than each carrying its own date stamp. Useful for multi-day field work where the report's natural organization is by day. This option visually declutters the report while still presenting dates and times.

GPS. A picker for how location data appears on cards: *None* (the default — location is omitted from the report), *Coordinates* (raw coordinates in the user's preferred format, set under GPS in Settings as either Decimal Degrees or Degrees Minutes Seconds), *Address* (a reverse-geocoded address, with fallback to coordinates if the address cannot be resolved), or *Both*. Whether GPS data is available on a given card depends on whether the user enabled *Auto-Attach GPS* in Settings or manually added a location to that card.

Activity Cards. A picker for how state-change cards from activities appear in the report: *Include All Cards* (every state change is rendered as its own card, in chronological order), *Stop Card Only* (the default — only the stop card is rendered; the stop card carries the full state-change log internally), or *Exclude* (activity cards are omitted entirely). Whether elapsed time is shown alongside these cards is governed by the *Include Elapsed Time* toggle under Activity Timer in Settings (on by default).

Countdown Cards. The same picker as that for Activity Cards, applied to countdown timers, with the same three options.

Group. A picker offering *Per Card* (the default — the group label is shown on each card) or *Do Not Show*. The label of this picker in the UI reflects the user's chosen name for the group level (set under Terminology in Settings): a user who renamed Group to *Crew* sees a Crew picker here, in keeping with the renaming behavior described elsewhere. The picker appears only when groups exist in the user's data.

Subgroup. The corresponding picker for subgroups, with the same two options. Like the Group picker, its label reflects the user's chosen name for that level (also set under Terminology in Settings), and it appears only when subgroups exist in the user's data.

Attribution. Present when the user has set a Known As name in their Identity profile (in Settings). When all cards in the report were authored by the user, the picker offers *Per Card* (the Known As name appears on each card), *Below Subtitle* (the default — the Known As name appears on a single line below the report subtitle), or *Do Not Show*.

When the report includes cards authored by more than one user (because some were transferred from collaborators), the picker is replaced with an advisory and per-card attribution is forced. Every card carries the Known As name associated with it at the time of its creation. The reason multi-user reports cannot suppress attribution is that the authoring user's content is locked, and attribution travels with the record.

Images: Delivery. Three options: *In Report* (the default — images are embedded in the PDF, attached to their cards), *In Folder* (images are delivered as a separate folder alongside the report, leaving the report as a textual index), or *None* (images are omitted entirely).

Whichever delivery option is chosen, the relationship between an image and its source card is preserved. When images are embedded *In Report*, the linkage is direct. When images are delivered *In Folder*, each image's filename includes its Card ID, so the linkage holds even though the images now live outside the report. CtxLens treats the integrity of these relationships — between cards, their elements, and their images — as a structural commitment that is honored across all export choices, not a per-option behavior.

Include Card Data. Off by default. The body of every CtxLens report includes the full content of each card — group, subgroup, tags, text, GPS, timestamp — alongside the images. This control is concerned only with the report's image appendix, where each image is reproduced for inspection. When this control is on, the card metadata is repeated above the first corresponding image in the appendix, so a reader looking at an image there has its source context immediately in view. When off, the appendix shows only the images and their identifiers.

Stamp Image IDs. On by default. Each image is stamped with a unique identifier corresponding to its place in the report. Useful when the report references images by ID and the user wants the IDs visible on the images themselves.

Stamp Corner. Selects which corner of the image carries the stamp: Top Right (the default), Top Left, Bottom Right, or Bottom Left. Visible only when Stamp Image IDs is on.

Stamp Font Size. Small, Medium (the default), or Large. Visible only when Stamp Image IDs is on.

Quality. Three preset levels: *Current Resolution* (images are output at their captured resolution, capped at 2,048 pixels on the longest dimension; suitable for archival fidelity), *Optimized for Print* (preserves the captured resolution at higher fidelity, producing larger files; suitable for printing), or *Optimized for Digital* (the default — capped at 1,200 pixels with more aggressive compression, producing smaller files; suitable for digital sharing). The starting point is the resolution at which the image was originally captured, set by *Max Dimension* under Image Capture in Settings (1,800 pixels by default).

What the user receives

A structured document, formatted for human reading, comprising the title (if titled), the cards in chronological order (or grouped by date when *Date-Grouped* is on), with images embedded or delivered separately according to the user's selection.

After generation, the user taps *Preview / Edit* to open the document in CtxLens's in-app editor. The document is fully editable there: the user can adjust headings, fix typos, refine wording, rearrange sections, or apply any other change the editor supports. Once any edits are saved, the user can export the result as a DOCX or PDF file. (For reports larger than 50 MB, the DOCX option is suppressed: DOCX files above that size become unstable, which is a property of the format itself, not a CtxLens-specific limitation.)

The output is fit for direct delivery to a client, a regulatory body, an internal archive, or any other recipient that wants a clean record of field work. It has not been seen or touched by AI.

5. The second path: on-device AI atop a standard report

The second path is for users who want the productivity benefit of an AI-generated summary at the top of their report but cannot, or will not, send their data to an external AI service. Common reasons include corporate policy that prohibits external AI use, contractual confidentiality obligations, or a personal preference for keeping work local. CtxLens supports this case by routing summarization through Apple Intelligence, which performs the work entirely on the user's device.

How the path is selected

In Report Settings, the user leaves the *Report Optimized for use with External AI* toggle off, and turns the *AI Summary* toggle on. The AI Summary section is visible in Report Settings only on devices where Apple Intelligence is available; on devices without it, the toggle simply does not appear, and the user proceeds along the first path. All other report controls remain available and behave exactly as described in the previous section. The AI Summary is additive: it sits on top of an otherwise normal report, governed by the same controls.

The AI summary uses the report's existing image numbering: each summary point that references a captured image carries the corresponding Image ID inline. A reader who was not on the scene can therefore see exactly which photographs back which observations and can quickly navigate from a point in the summary directly to the image(s) cited. This makes the summary less an abstraction over the work and more a guided tour through it.

When a card contains more than one image, the summary indicates the count alongside the Card ID, and each photograph carries an “(N of M)” designation on its stamp — for example, “7 (2 of 4)”. The reader can see, at a glance, both the source card and the specific photograph within it.

What the user controls (in addition to the previous section)

When AI Summary is on, three additional controls appear.

Summary Type. A segmented control selecting the form of the summary: Narrative (continuous prose, the default) or Bullets (a structured list). The user chooses based on how the summary will be read.

Word Count (max). A slider that sets the approximate target length of the summary, from 200 to 1,000 words in steps of 50, with a default of 300. The label notes that the actual length may vary; the slider sets a ceiling, not a precise count. In practice, Apple Intelligence does not always honor that ceiling. CtxLens applies several programmatic techniques to hold the model to the user's target, and the techniques work most of the time, but the underlying model still overshoots from time to time. The slider should be read as a target, not a guarantee.

Additional Instructions. A sub-screen where the user can add specific guidance to the summarization model — for example, a directive to focus on safety observations or to emphasize a particular type of finding. The instructions are passed to Apple Intelligence as part of the summarization request.

A label beneath the AI Summary controls is worth quoting verbatim: *“Uses Apple Intelligence to generate a summary. Summarization is performed entirely on your device.”* This is the contract. No card content is transmitted off the device during summarization. The summary is generated locally by the on-device model and inserted into the report.

What the user receives

The same structured document a first-path user receives, with one addition: an AI-generated summary of the selected card population is placed at the top of the report, above any other content. The summary's form (narrative or bullets), its length (within the user-set ceiling), and its guidance (from Additional Instructions) reflect the user's choices. Everything else in the report — the cards, their formatting, the images, the attributions — follows the controls described in the previous section. The preview, edit, and export flow described in the previous section applies here unchanged: the document opens in the in-app editor, edits can be made and saved, and the result is exported as DOCX or PDF (DOCX suppressed above 50 MB).

The output is fit for the same recipients as the first path, with the added benefit that the recipient sees a summary up front. Because summarization happens on-device, the user has not exposed any card content to an external service in producing it.

6. The third path: external AI

The third path is for users who intend to take the report's output and use it as context for an external AI (typically a multi-modal, third-party large language model) to produce a complex deliverable, or to populate reporting templates. The deliverable might be a formal report drafted from the cards, a populated client-supplied template, an investigation summary, an appraisal narrative, a daily report rendered into a custom format or template, or any of the many shapes that AI is well-suited to generate from well-structured input. The argument for why this matters is the subject of *On Context*, a foundational essay. A distillation is included as an appendix to this document; the full essay may be requested by email at support@ctxlens.com.

How the path is selected

In Report Settings, the user turns on the *Report Optimized for use with External AI* toggle. This is a single decision and it is exclusive. When the toggle is on:

- All other report controls in the view become inoperable.
- Any choices the user made before turning the toggle on are suppressed.

The exclusivity is mechanical, not advisory. The user cannot, for example, generate an external-AI-optimized report that also includes an on-device AI summary, or one that uses the human-reading formatting controls. The toggle either replaces the rest of the configuration or sits alongside nothing else.

The reason for the mechanical enforcement is the same reason the toggle exists at all: an export intended for machine reading is structurally different from an export intended for human reading, and combining them would compromise both. The decision must be unambiguous, and CtxLens enforces it.

What the user receives

A PDF formatted for ingestion by a multi-modal AI. (That is, a model that reads both the text of the document and the images embedded within it.) Sections are clearly delimited, cards are rendered with their full element set, tags are present and searchable, groups and subgroups provide an outline, images are embedded at machine-usable resolution with their relationships to cards preserved, and the text is plain. All relationships are preserved. Every formatting choice in the export is made in service of being parsed cleanly by an external model.

Unlike the first two paths, this output is delivered as a PDF only, and the in-app editor is not offered for it. After generation, the PDF opens automatically in a preview view, where the user can review it before choosing to share or save. There is no edit affordance for this format. The structural integrity of the export is what makes it useful to the model; allowing the user to alter that structure would risk compromising the very property the path was selected to produce.

What the user does next

The user takes this PDF and attaches it to a prompt for the external AI of their choice, on whatever terms (and under whatever authorizations) apply to their use of that AI. CtxLens does not call the external AI. CtxLens does not have an opinion about which external AI the user should use. The export is the handshake; what happens after the handshake is the user's choice.

The purpose of the path is straightforward: give the external AI enough structured, relational context that the deliverable arrives in one or two passes, instead of the many iterations a less-structured prompt typically requires. The user's time is the resource the export is engineered to save.

7. Closing note

The three paths are mutually exclusive at selection time and mutually distinguishable at delivery time. A user who chose no AI receives an output that has not been seen or touched by AI, in any form. A user who chose on-device AI receives an output whose AI processing happened on their device. A user who chose external AI receives an output formatted for an external model and makes the explicit decision to attach it to one.

The product enforces these boundaries because the consequences of conflating them are real. A user who believed they had kept their data local but had inadvertently sent it to an external service may be in breach of policy or contract. A user who believed they had received a polished human-readable report but had instead received a machine-optimized one would not have the document they expected. CtxLens removes the possibility of either by making the choice explicit, exclusive, and resistant to drift in the moments leading up to the report being generated.

This is a descriptive document. It is meant to clarify what CtxLens offers, not to argue that one path is preferable to another. In all cases, and at all times, the user, knowing their own work and their own constraints, decides which way to proceed.

Appendix A: A Distillation of On Context

Distilled from On Context · April 2026

"The best AI prompts will not be written. They will be captured, as a byproduct of work."

THE ASYMMETRY

The AI industry has spent three years improving what models can do with context, and almost none on how humans *produce* it. Models have gotten dramatically better at reading; humans have not gotten meaningfully better at writing what models need. That asymmetry is why so many enterprise pilots stall, why roughly ninety percent of CEOs report no productivity gains, and why casual users walk away disappointed. The model side is doing its job. The human side has not had a tool built for this job.

WHY CURRENT APPROACHES MISS

Wearables capture the eye but not the judgment behind it. When an inspector lingers four seconds on one weld out of forty, the recording shows the weld — not the reason. Document-dumping into chat is the same posture without the camera: bytes in, judgment expected back. Both share an assumption worth rejecting: that the user's job is to accumulate inputs and the model's job is to extract meaning. The opposite is closer to true. The user's job is to *produce* meaning by deciding what counts. You cannot prompt-engineer your way out of this — by the time you are at a keyboard reconstructing the scene, the scene is already gone.

CAPTURE IS JUDGMENT, NOT RECORDING

Capture, in the sense that matters, is the act of deciding,while the work is happening, what is worth capturing. That decision is judgment, and judgment is the one thing a model cannot supply. If it is not captured when it exists, it is lost. The fix is not a bigger or better model. It is a tool that makes deliberate capture efficient enough to become a habit.

CTXLENS, IN ONE PARAGRAPH

CtxLens is a native iOS app whose atomic unit is the **card** — one moment of captured context, with optional images, a user-defined group and subgroup, tags, GPS, timestamp, and up to 1,000 characters of text. The constraints are commitments. **No document attachments**: offloading is not capture. **Capped text fields**: structure is the rule, writing the exception. **User-built taxonomies**: the user's mental model of the work, made visible — and inherited by every export. **Transferable cards with locked authorship**: authored judgment is immutable;

surrounding context accumulates, attributed. The output is a single PDF export engineered for clean machine consumption.

THE RESULT

Complex deliverables that would have taken many iterations from a stream-of-consciousness prompt (with or without attachments) can, with a well-structured export attached, be produced in one or two. Not because the model is better. Because the *input* is.