Send Codex the pages that matter.

We tested Token Saver on long brand guides, research reports, and similar PDFs. It searches locally and gives Codex a few page-cited passages instead of every page.

88.9% less text
: returned from the PDFs on average

15 of 15
: test questions found an accepted page in the top five results

5 public PDFs
: from 10 to 492 pages in one local test

The decision: use Token Saver for long, text-heavy PDFs and repeated focused questions. Read short documents directly. Use visual inspection for scans, diagrams, and layout. We measured less PDF text sent to Codex, not lower total cost or better answers.

See the business value See the evidence Use the decision guide Inspect the implementation

What this changes for a video team

A video project often starts with documents rather than footage. A team may need to find the approved tone in a brand guide, a required disclaimer in a legal brief, or the target audience in a campaign plan.

Codex does not need every page to answer each of those questions. Token Saver keeps the full PDF outside the conversation, searches it on the local machine, and gives Codex a small set of passages with page numbers.

This has three practical benefits:

- **More focused answers.** Less unrelated text competes with the instructions and evidence needed for the current question.
- **Easier review.** Page references help a person check the source before a production decision is accepted.
- **Reuse across questions.** A long document can be indexed once and searched again when the team asks a follow-up question.

These are operational benefits, not proven financial savings. We have not yet shown that the complete Codex session costs less, finishes faster, or produces a better final answer.

What the test showed

We tested a reviewed version of the open-source Token Saver repository on five public PDFs from official US government sources:

- Copyright Basics
- NIST AI Risk Management Framework 1.0
- the Dobbs opinion
- NIST SP 800-53 Revision 5
- NASA Systems Engineering Handbook

The documents ranged from 10 to 492 pages and from about 5,184 to 332,621 extracted tokens, a rough measure of text size. We wrote 15 questions whose answers were tied to known source pages. The NASA handbook was held back until a later check so that it could test whether the approach worked on a new document.

Search method	Accepted page found in the top five
Exact-word search	13 of 15 questions
Exact words plus related meaning	15 of 15 questions

Across the 15 questions, Token Saver returned **88.9% less PDF text on average** than sending all extracted text. On the three largest documents, it returned about **98.0% to 99.2% less**.

This supports one limited finding:

> In this test, Codex received much less PDF text while an accepted source page still appeared in the first five search results for every question.

It does not show that every answer was correct. It does not show that the same result will hold for any PDF or question. It also does not show lower total Codex usage, cost, or time.

Less PDF text is not the same as a cheaper agent

The PDF is only one part of a Codex session. The system may also load project instructions, conversation history, tool descriptions, earlier results, and the model's own response.

Our first comparisons did not keep those other conditions equal. One control did not answer the same PDF question. Another used a different set of tools and permissions. The Token Saver run returned less PDF text, but its complete record was larger because more supporting machinery was loaded.

That observation does not show that Token Saver caused higher use. It shows why the comparison was not fair.

A fair test must keep the model, question, tools, permissions, and workflow the same. It should change only one thing: whether Codex receives the complete PDF text or the smaller set of retrieved passages. Building the index for the first time should also be measured separately from reusing an existing index.

Until that test is complete, the safe conclusion is simple:

> Token Saver reduced the PDF text given to Codex in our test. Its effect on the cost and speed of the whole workflow is still unknown.

Search results are clues, not proof

Document search can return a passage that sounds relevant without supporting the claim in the question.

We tested this by asking 10 questions that the selected documents did not answer. The original Token Saver server still returned passages for all 10. Simple checks rejected many obvious mismatches, but some false questions used the same names and terms as the document and still looked convincing.

This is why a similarity score cannot be treated as truth. A court opinion may mention the right law and person but quote a dissent rather than the majority. A brand guide may mention a color without approving it for the requested use.

Our Codex workflow therefore follows five steps:

1. Register the PDF.
2. Ask one focused question.
3. Read the returned passages.
4. Check whether they cover the important parts of the question.

5. Answer with page references and state what remains uncertain.

The checks can reveal missing terms or weak coverage. They cannot prove that a passage is complete, correctly attributed, legally binding, or sufficient for a high-risk decision. A person or agent still has to read the evidence.

Choose the right way to read each PDF

The goal is to notice every PDF and choose the right reading method. The goal is not to force every PDF through Token Saver.

PDF and task	Best starting method
Long, text-heavy PDF with repeated focused questions	Token Saver search
Short, text-only PDF	Read it directly
Scanned or image-only PDF	Read the text with OCR and inspect the pages
Diagram, typography, or layout review	Inspect the full pages visually
Whole-document summary	Read every section with a completeness check
Proving that something is absent	Search broadly and state exactly what was checked

We tested the difficult edge with six fresh Codex runs about a short, scanned, diagram-heavy brochure. All six chose visual inspection first. None made Token Saver the main method. This supports the table for that repeated task, but it is not a general success rate for every model, machine, or document type.

Where it fits in Instavar

Token Saver can help during planning. It can find textual requirements such as:

- the approved tone
- the target audience
- a required disclaimer
- a prohibited phrase
- a campaign objective

Those findings can become inputs to a script or production brief. They cannot approve the finished video.

A page reference does not prove that:

- an image, voice, or clip is licensed for the customer
- the final composition follows the visual brand rules
- a crop is safe for a vertical platform
- captions and narration match the final timeline
- the rendered video passed picture and sound review
- the selected social account is authorized
- a person approved the final publish action

The Codex plugin is therefore a local research and operator tool. It is not a production feature inside Instavar Studio. A production version would also need customer separation, source versions, access rules, index refresh and deletion, and the existing render, review, and publishing controls.

Privacy: what stays local and what may leave the machine

The full PDF can stay on the local machine while selected passages are sent to a cloud-hosted model. Those are different claims.

Local text extraction and local search reduce how much document text is sent to the model provider. They do not guarantee that no document content leaves the machine. The saved search index also contains document text and numerical search data, so it needs the same care, retention period, and deletion rules as the source PDF.

Folder allowlists, file checks, size limits, and source fingerprints reduce risk. They do not replace the user's permission or the controls needed to keep one customer's information separate from another customer's information.

We used only public, non-private PDFs in this evaluation.

How the Codex integration works

The main idea is simple:

PDF on the local machine
-> extract and divide its text into small passages
-> search for exact words and related meaning
-> remove repeated results
-> send a few page-cited passages to Codex

We packaged the integration as a Codex plugin. A plugin is the container that groups the parts needed to install and use the feature. Our package contains:

- a short instruction that tells Codex when to use PDF search
- a prompt check that notices visible PDF references
- a safety check before supported local PDF-reading commands run
- a local server that registers, searches, and checks PDF passages

OpenAI's documentation explains the wider Codex plugin and hook systems.

The prompt check runs whenever a prompt is submitted, but it does not add text to every conversation. It inspects the prompt and exits silently when there is no visible PDF signal. When a PDF is involved, it adds the PDF reading rules.

In 100 local runs using ordinary prompts with no PDF, the check added no text to the model's prompt. Its middle run time was 27.19 milliseconds on this Mac. Turning on optional measurements added about 0.65 milliseconds to that middle value.

This timing applies only to the tested Mac, script, and prompt set. It does not prove the same speed on other machines or under heavy parallel use. Local hooks also do not cover every possible hosted tool path, so they are a useful routing check rather than a complete security wall.

The decision and the next test

We kept Token Saver as a selective local PDF reader for Codex.

Use it when the document is long and text-heavy, the question is focused, and several questions can reuse the same index. Avoid making it the default when the document is short, scanned, visual, or must be read in full.

The broader lesson is:

> Sending less text helps only when the smaller selection still contains everything needed for the decision.

A large reduction has negative value if it removes an exception, diagram, speaker, date, or approval rule.

The next useful comparison will keep the full Codex environment fixed and change only how the PDF text is supplied. It should measure total usage, time, answer support, correction work, and human review. We also need new tests with scanned, multilingual, table-heavy, damaged, oversized, and multi-document cases.

For now, the supported conclusion remains narrow: local search can greatly reduce the PDF text given to Codex for large text documents and focused questions. Whether that improves the complete agent workflow still needs to be measured.

If your video workflow begins with long research, brand, or compliance documents, decide what evidence each production decision needs before choosing how the documents will be searched.

Download the printable version . Last updated 1 Aug 2026. This article reports one reviewed Token Saver version, five public PDFs, 15 written questions, one macOS Codex environment, and follow-up routing checks. Do not apply the results more broadly without further testing.